ANNO 3 N. 7

MICROPROCESSORS-HARDWARE - SOFTWARE
HOME & PERSONAL COMPUTERS

L. 2000 *

DATA BASE PERSONALE

NUOVA RUBRICA
"COMPUTER PRATICO"

Spedizione in abb. postale Gruppo III/70

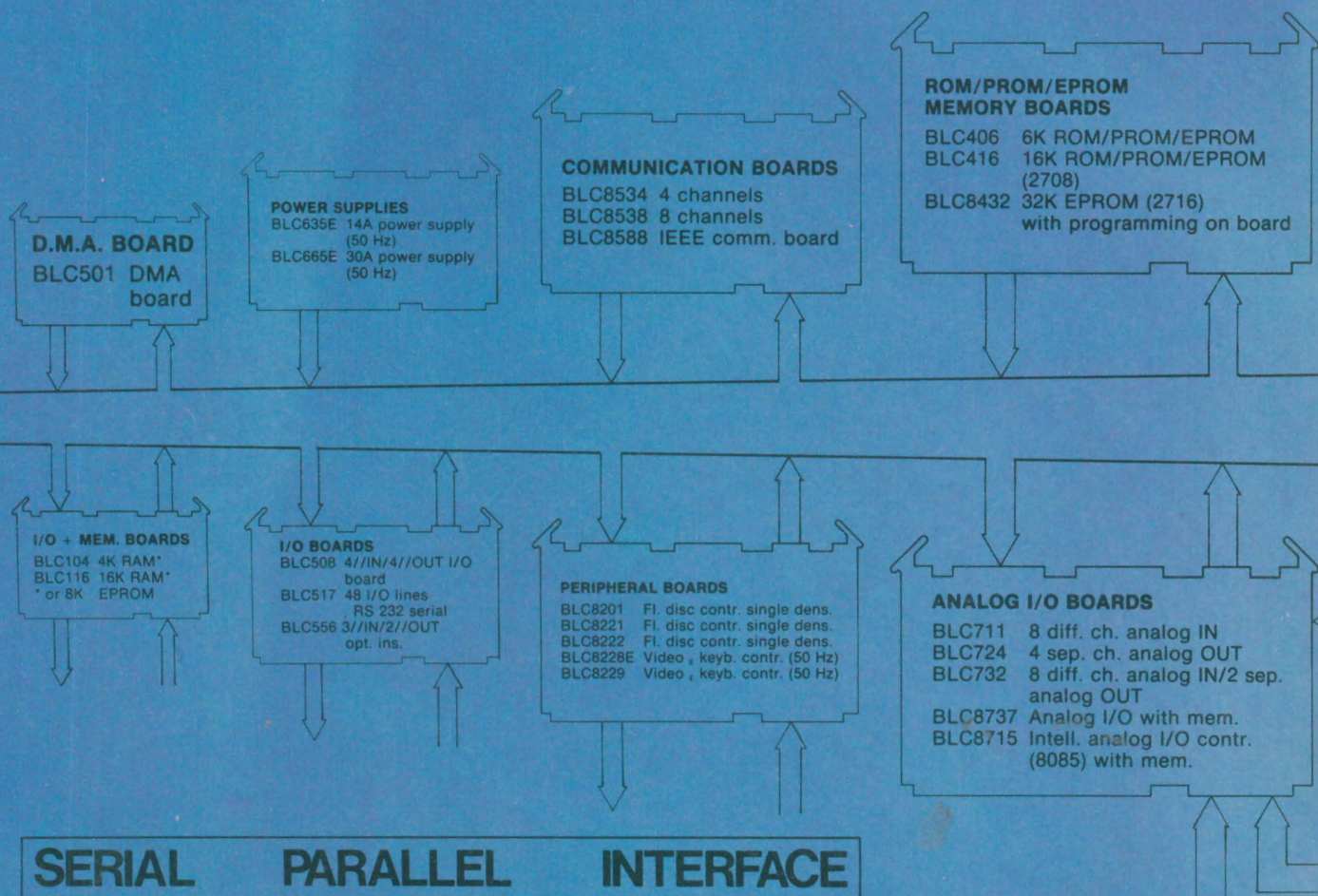
APPLICAZIONI GESTIONALI:
I COSTI AZIENDALI

PICOCOMPUTER: INTERFACCIAMENTO
CON REGISTRATORE A CASSETTE

CORSO SUL PASCAL

IL CALCOLATORE PER I GIOVANI

TRASMISSIONE SERIALE

Serie 80 BLC microcomputers

La famiglia dei prodotti della Serie/80 della National Semiconductor offre una base completa di prodotti per microcomputer. Completamente compatibili con l'architettura della Serie/80 per l'industria, molti di questi prodotti possono sostituire plug-for-plug i prodotti SBC.
La serie BLC National pur mantenendo con essa la compatibilità sia hardware che software, è caratterizzata da un progetto originale, da un elevato numero di opzioni per una grande flessibilità d'impiego e dalla garanzia totale per un anno.
Oltre ad essere seconda sorgente per alcuni prodotti, la National ha sviluppato anche prodotti totalmente nuovi, nell'intento di soddisfare sempre più le richieste del mercato. Il Sistema di Sviluppo STARPLEX è un esempio di risposta della National alle esigenze dell'industria.

Bit

SOMMARIO

Anno 3 - N° 7 - Aprile/Maggio 1980

In copertina:
Il computer è fantastico
Grafica di Renato NISI.

INDICE INSERZIONISTI

DIRETTORE RESPONSABILE
Giampietro Zanga

COORDINATORE TECNICO
Marcello Montedoro

CAPO REDATTORE
Dino Bortolossi

COORDINAMENTO PERSONAL COMPUTER
Gloriano Rossi

SEGRETARIA DI REDAZIONE
Cecilia De Serio

GRAFICA E IMPAGINAZIONE
Job Line srl

DIFFUSIONE E ABBONAMENTI
Gabriella Napoli, Silvia Decari
Marco Benedetti

DIREZIONE, REDAZIONE
P.le Massari, 22 - 20125 Milano
Telefoni 68.03.68 - 68.00.54

AMMINISTRAZIONE
Via Vincenzo Monti, 15 - 20123 Milano

PUBBLICITA': Concessionario per l'Italia e l'Estero Reina & C. S.n.c.
Via Ricasoli, 2 - 20121 Milano
Tel. (02) 803101-866192-8050977
864.066 - Telex: 320419 BRUS I

Stampa: Tipo-Lito Elcograf s.p.a.
22050 Beverate (Como)

Concessionario esclusivo per la diffusione in Italia e all'Estero:
SODIP - Via Zuretti, 25 - 20125 Milano

Spedizione in abbonamento Postale Gruppo III/70

Prezzo della rivista L. 2.000

Numero arretrato L. 3.000
Abbonamento annuo L. 12.500
per l'Estero L. 17.500

I versamenti vanno indirizzati a:
Jackson Italiana Editrice S.r.l.
Via V. Monti, 15 - 20123 Milano
mediante emissione di assegno bancario, cartolina vaglia o utilizzando il c/c Postale numero 11666203
Per i cambi di indirizzo, indicare, oltre naturalmente al nuovo, anche l'indirizzo precedente, ed allegare alla comunicazione l'importo di L. 500, anche in francobolli.

NANOCOMPUTER®

UN COMPUTER PER IMPARARE TUTTO SUI COMPUTER.

In questi ultimi anni, l'eccezionale diffusione dei microprocessori nell'industria e nella vita di tutti i giorni ha aumentato fortemente la richiesta di persone in grado di operare professionalmente nel settore.

La SGS-ATES, uno dei maggiori produttori di microprocessori da sempre in primo piano nel loro supporto in Europa, ha fatto fronte a questa esigenza realizzando il NANOCOMPUTER, un sistema didattico professionale e completo.

NBZ80-S. Scheda base, scheda per esperimenti, miniterminale, contenitore-alimentatore, kit di fili, Nanobook 1 e 3, manuale tecnico.

<u>Insegnamento e apprendimento: due facce dello stesso problema.</u>

Su questo concetto è basato il sistema didattico NANOCOMPUTER in cui la SGS-ATES ha riversato una lunga esperienza sistemistica e produttiva, realizzata preparando i suoi tecnici e ricercatori ad altissimo livello.

Il NANOCOMPUTER è un sistema didattico integrato e modulare. È formato da un potente microcalcolatore con il microprocessore Z80 prodotto in Italia dalla SGS-ATES, e da un insieme completo di sussidi educativi: libri di testo Nanobook® in italiano e nelle principali lingue europee, manuali tecnici, kit per esperimenti.

La concezione modulare permette al NANOCOMPUTER di crescere insieme allo studente, in un processo di apprendimento attivo fondato sul continuo dialogo tra la macchina e lo studente.

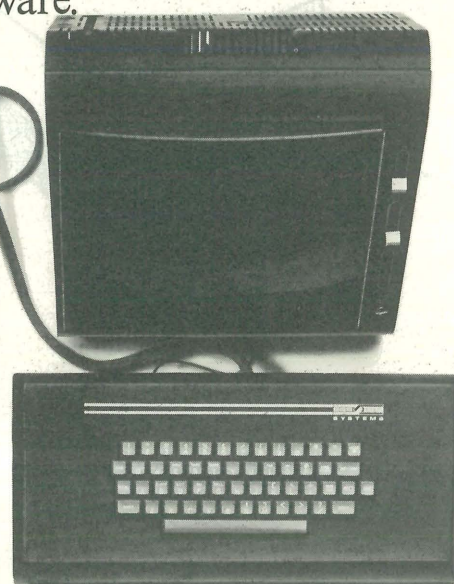
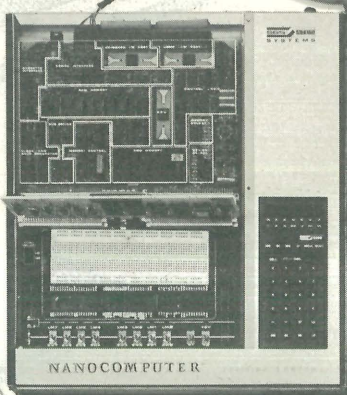
SGS-ATES Componenti Elettronici S p A , Via C. Olivetti 2, 20041 Agrate Brianza, Tel. (039) 65551 • DIREZIONE COMMERCIALE ITALIA, Via Correggio 1/3, 20149 Milano, Tel. (02) 4695651 • UFFICI VENDITA: Firenze, Tel. (055) 4377763 • Milano, Tel. (02) 4695651 • Roma, Tel. (06) 8392848 • Torino, Tel. (011) 531167 • PUNTI DI VENDITA: Via Larga, ang. Via Brolo/Via Verziere, 20122 Milano, Tel. (02) 8690047 • Via S.

Per queste caratteristiche, il sistema NANOCOMPUTER è particolarmente adatto non solo all'apprendimento a scuola, sotto la guida di un insegnante, ma anche per chi voglia individualmente prepararsi a questa nuova professione.

NBZ80-B. Scheda base, miniterminale, contenitore-alimentatore, Nanobook 1, manuale tecnico.

Il sistema NANOCOMPUTER: un sistema modulare. Il NANOCOMPUTER, studiato espressamente per impieghi didattici, riunisce in sé un'elevata rigorosità di concezione e un'estrema flessibilità, essenziali in un processo di apprendimento teorico e sperimentale al contempo. Nella sua versione più semplice, NBZ80-B, il NANOCOMPUTER permette anche allo studente senza conoscenze specifiche di impadronirsi delle tecniche di programmazione dei microprocessori.

Con la versione NBZ80-S lo studente viene introdotto anche nelle tecniche di interfacciamento di un microprocessore con il mondo esterno e nei problemi di interazione tra hardware e software.

È possibile, attraverso un kit di espansione, passare dalla versione NBZ80-B alla NBZ80-S. In tal modo ogni studente può scegliere, graduandolo nel tempo, il livello di apprendimento più consono alle proprie esigenze. L'NBZ80-S è a sua volta ulteriormente espandibile per consentire l'approfondimento di un linguaggio ad alto livello, il Basic, soprattutto nelle sue interazioni con l'hardware.

NBZ80-HL. Con 16K byte di RAM, tastiera alfanumerica con interfaccia video, 8K ROM di Basic su scheda addizionale, libro Basic Programming Primer, monitor TV (opzionale).

Desidero ricevere gratuitamente maggiori informazioni sul sistema NANOCOMPUTER®

NOME ____________ COGNOME ____________

INDIRIZZO ____________

PROFESSIONE ____________

Inviare a: SGS-ATES
Componenti Elettronici S p A
Via C. Olivetti 2, - 20041
Agrate Brianza, tel. (039) 65551

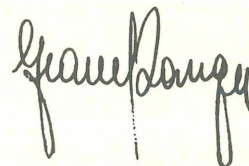
SGS ATES

Bit

Quintino 29/C, 10121 Torino, Tel. (011) 531267 • DISTRIBUTORI: CID, Roma, Tel. (06) 6383979 • DE DO Electronic Fitting & C. S.a.s., Tortoreto Lido (Te), Tel. (0861) 78134 • DISELCO S.p.A., Milano, Tel. (02) 3086141 • FANTON ELECTRONIC SYSTEM S.r.l., Padova, Tel. (049) 654487 • FANTON BOLOGNA S.r.l., Bologna, Tel. (051) 357300 • AGENTE ESCLUSIVO PER LA SCUOLA: GB PARAVIA & C. EDITORI, Torino, Tel. (011) 779166.

EDITORIALE

Pedalare a tutta forza!

Va bene, va bene, il famigerato "Non leggete qui sotto" era una provocazione, lo confesso, l'ho fatto apposta! Ma voi mi avete preso sul serio (grazie, i miei redattori ogni tanto non lo fanno) e continuate a bombardarci di idee, critiche, pareri, proposte e suggerimenti. Insomma, non si può più dormire, in redazione!!!

Dalle vostre lettere, oltre ad una buona quantità di idee interessanti, sta emergendo un fatto che ci rende quasi facili: abbiamo fatto centro. Quando siamo partiti la direzione che intendevamo prendere, il nostro obiettivo era una rivista che potesse interessare molti tipi diversi di lettori, a livelli diversi di preparazione e di professionalità, per intenderci, dall'hobbista al ricercatore professionale, dall'appassionato all'analista. Ebbene, le vostre lettere provengono forse da una fascia più vasta ancora.

C'è però qualche cosa che avete in comune: intanto una discreta preparazione sull'argomento. Non credo una parola di quelli che si professano piuttosto ignoranti per poi tirar fuori degli argomenti su cui i nostri esperti si scatenano in feroci discussioni, il vero ignorante normalmente viene liquidato con due paroline bonarie ed un sorrisetto.

Poi avete una gran passione per il "pensiero elettronico", qualsiasi cosa questo strano modo di dire significhi. Devo ricordarmi di chiedere al medico se l'entusiasmo in forti dosi può far male, di certo il vostro entusiasmo è contagioso e ce lo siamo preso tutti, anche chi si riteneva vaccinato.

Poi ancora siete quasi tutti giovani: molti anche anagraficamente (beati loro), tutti nell'animo. Chi è vecchio non si lascia entusiasmare, diventa cinico e sprezza le novità. Ma fra le vostre lettere non c'era l'ombra di cinismo, c'era invece in quasi tutte una gran voglia di fare, di trovare e di costruire novità, di capire e di collaborare.

Quindi non ci rimane che accontentarvi, o almeno pedalare a tutta forza per cercare di farlo nel migliore dei modi.

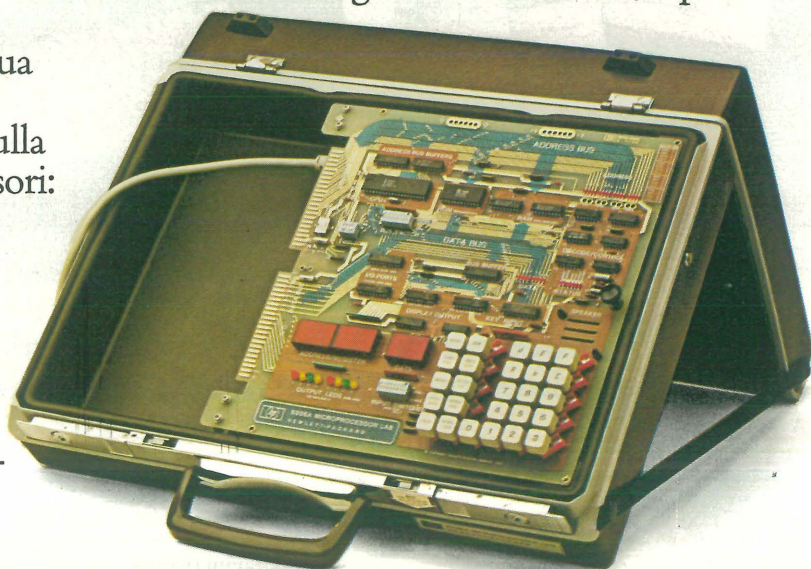
Praticamente unico.

Finalmente un testo pratico che serve davvero a mettere le mani sui Sistemi a microprocessore.

Fino ad oggi, i libri di testo sui microprocessori erano più che altro dedicati ai progettisti, ed erano molto teorici.

Ecco, invece, un manuale essenzialmente pratico, in lingua italiana, che insegna tutto sull'hardware, sul software e sulla ricerca guasti nei microprocessori: sono circa 460 pagine che comprendono 20 lezioni complete di introduzioni, riassunti e quiz pratici per meglio memorizzare le nozioni.

In più, le appendici contengono tutta la documentazione sia di hardware che di software necessaria.

Il libro è curato dalla Hewlett-Packard, di cui segnaliamo qui tra l'altro il laboratorio portatile 5036A, una valigetta completa di microcomputer e alimentatore, espressamente ideata per eseguire gli esperimenti che si susseguono nel volume, e per l'addestramento alla ricerca guasti nei Sistemi a microprocessore.

TAGLIANDO D'ORDINE, da inviare a:
Jackson Italiana Editrice,
p.le Massari 22, 20125 Milano

Inviatemi N° copie del volume: "Practical Microprocessors: hardware, software e ricerca guasti", al prezzo di Lit. 35.000 cad. più le spese di spedizione.
☐ pagherò al postino.
☐ allego assegno (in questo caso la spedizione è gratuita).

Nome Cognome ____________________

Posizione ____________________

Ditta ____________________

Codice fiscale (per le ditte) ____________________

Via ____________________

Città ____________________ CAP ________

BIT

È tempo di BIT

È tempo di BIT '80, ovvero è tempo per analizzare che cosa è avvenuto in Italia, a distanza di un anno, nel campo della elaborazione personale. In attesa della seconda rassegna del personal computer (che si svolgerà a Milano dall'8 all'11 maggio), dopo la quale sarà possibile fare considerazioni più aggiornate anche sulla base di quanto verrà detto nell'ambito del convegno che sarà tenuto parallelamente alla rassegna, è possibile già evidenziare alcuni fatti significativi.

Il primo elemento da annotare è il maggior numero di prodotti destinati all'utenza "personale": sono comparsi marchi già affermatisi in altri paesi (un esempio per tutti è l'Apple); nomi prestigiosi (quali SGS-ATES, Siemens, Texas Instruments, Hewlett-Packard) sono scesi in un campo che finora aveva visto assenti i grandi nomi dell'elettronica mondiale; sono stati presentati da parte di marchi già noti prodotti più sofisticati (come nel caso di Radio Shack e PET Commodore). Ad una prima impressione sembrerebbe che il maggior numero di prodotti contribuisca ad aumentare l'incertezza della scelta; ma a ben vedere si può affermare il contrario: i differenti prodotti oggi presenti sul mercato italiano rispondono a differenti esigenze di utilizzo.

Se inizialmente (circa un anno e mezzo fa) i personal computers presenti avevano per la maggio parte caratteristiche "strutturali" abbastanza omogenee a fronte delle quali venivano offerte prestazioni assurde, oggi almeno c'è chiarezza su questo punto: la differenza tra i vari prodotti, concretamente valutabile in termini di Mlire, evidenzia una differenza di prestazioni non giustificabile diversamente.

La differenzazione più marcata è da segnalare nel campo dei prodotti destinati all'utenza professionale-personale, tanto che si può parlare di "seconda generazione". A differenza dei sistemi della "prima generazione", la cui validità, per la maggior parte, era e rimane circoscritta al campo delle applicazioni hobbistiche piuttosto che a quello delle applicazioni professionali, i nuovi sistemi sono caratterizzati da un "vestito" che evidenzia il salto di qualità di questi prodotti e la tendenza dei vari costruttori a muoversi verso quell'area di mercato che, per ragioni oggettive, e più sensibile all'elaborazione elettronica.

In ultima analisi, oggi vi è un maggio numero di prodotti sofisticati: la presenza costante di memorie di massa quali mini-floppy o floppy-disk, sistemi operativi aderenti a quanto solitamente s'intende con questo termine (tanto per citare alcuni elementi strutturali qualificanti) permettono di definire, al di là di ogni discorso, la linea di demarcazione tra i vari tipi di prodotti e quindi di prestazioni.

È chiaro che l'aumento, verificatosi in così breve tempo, del numero di sistemi ora disponibili può porre interrogativi su come l'utente possa essere stato influenzato nel passato. Premesso che questo fatto è stato amplificato dal ritardo con cui l'elaborazione personale è stata "scoperta" in Italia, e che quindi questo periodo è stato un periodo di adeguamento del mercato italiano alla realtà merceologica creatasi nel frattempo (soprattutto oltreoceano), sono necessarie ulteriori considerazioni.

Posto che è sempre opportuno che l'utilizzatore abbia un'idea abbastanza precisa di quello che una macchina può o non può fare, e di quello che vuole, il discorso inevitabilmente porta alla politica commerciale degli operatori intervenuti in questo settore. A quesò proposito un'unica osservazione: inizialmente si sono viste iniziative improntate ad un approccio troppo riduttivo per poter credere che su queste basi potesse svilupparsi da zero l'elaborazione personale. La presenza trainante di aziende che hanno operato notevoli investimenti in questo settore è oggi un elemento che contribuisce a dare maggiore credibilità e quindi penetrazione alle applicazioni del computer nell'area personale-professionale.

Per concludere, è tempo, come si diceva all'inizio, di tirare le prime somme. Per questo motivo BIT promuove un sondaggio nell'area dell'elaborazione personale avente il fine di valutare le sue caratteristiche, le attese e/o le delusioni, in una parola l'utilizzo o il non utilizzo del microelaboratore.

Ovviamente sottoponiamo ad indagine anche il nostro operato, per sapere in che misura abbiamo contribuito, nel nostro piccolo, a creare una conoscenza informatica.

NEWSLETTER

Un club di microinformatica a Torino

Chi risiede nell'area torinese e desidera iniziarsi alle tecniche della microinformatica può rivolgersi ad un club in corso di formazione. La microassociazione si chiama MICROTEL (non ha nessuna parentela con gli omonimi organismi francesi) e come corredo iniziale porta un Apple II più, tra breve, dei sistemi in kit. Fra i promotori dell'iniziativa figura il sig. Vittorio Mascarello al quale gli interessati possono, in questa fase di transitorietà rivolgersi per ogni informazione. L'indirizzo è: Corso Cosenza, 85, Torino (Tel. 3093014).

Il computer si compra in COMPUTERIA

A Milano in Via della Moscova, 23, la COMPUTERIA ha aperto un nuovo negozio per la vendita al pubblico di piccoli sistemi per la elaborazione elettronica dei dati. La "COMPUTERIA" di Milano (il nome è stato brevettato dalla Unicomp) è solo la prima di una serie. La Unicomp sta infatti lanciando un programma di "franchising" che, nelle aspettative dei suoi dirigenti, dovrebbe presto provocare un fiorire di negozi in varie località italiane.
"La nostra proposta — dice l'ing. Sergio Focardi della Unicomp — si indirizza a quelli investitori privati che possono essere interessati a giocare la carta del personal computer per le enormi prospettive che il settore promette negli anni '80, e che hanno bisogno di un'organizzazione che permetta loro di partire col piede giusto. Noi diamo ai nostri negozi con questo piano di franchising l'utilizzo di un nome che comincia ad essere conosciuto e apprezzato, la credibilità commerciale e tecnica della nostra organizzazione, la disponibilità di prodotti che sono cavalli vincenti".
Il progetto di franchising è un naturale arricchimento dell'attività della Unicomp, che già opera sul mercato italiano attraverso una rete di rivenditori ("dealers" secondo la corrente terminologia americanizzante).

La Triumph si affaccia al personal computing

Con l'approssimarsi dell'estate vanno chiarendosi gli obiettivi della Triumph-Adler sul mercato della elaborazione personale. Da quando la Volkswagen ha assunto il controllo nel costruttore tedesco di macchine per ufficio, è forse questa la prima grossa novità. La Triumph-Adler entra sul mercato del personal computing con la serie di apparecchi Alphatronic. Si tratta di tre modelli dai costi compresi (le valutazioni sono indicative) fra 700.000 e 3.700.000 lire.
L'unità più piccola ha una ROM di 12 Kbytes, una RAM di 4 Kbytes, una cassetta ed una monitor TV.
La versione di mezzo consta di 16 Kbytes di memoria e di due minidischi. Il sistema più grosso, oltre ai due minidischi e ad una memoria centrale di 48 Kbytes, è dotato di una stampante da 2200 caratteri al secondo. Senza il sostegno finanziario della Volkswagen la Triumph-Adler non avrebbe quasi sicuramente tentato la carta del personal computer e non sarebbe neppure diventata un fornitore di stampanti per il mercato OEM.

Una fabbrica italiana di personal computers?

La General Processor di Firenze sta concretamente valutando la possibilità di costruire uno stabilimento nel Mezzogiorno. Il progetto presupporrebbe un salto quantitativo considerevole per la piccola azienda fiorentina nata quasi per divertimento circa tre anni fa. La General Processor, formata da soci-progettisti, ritiene di aver trovato l'asso giusto per sfondare sul mercato della elaborazione personale. Si tratta del Sistema T, un apparecchio per applicazioni essenzialmente gestionali concepito con una filosofia da sistema general purpose, e con i prezzi varianti da poco più di due milioni a quasi 12 milioni, a seconda delle configurazioni. Si tratta del primo prodotto realizzato "in serie" (due sistemi completamente testati al giorno). Precedentemente la General Processor si era fatta conoscere con i sistemi CHILD 8BS e CHILD Z.

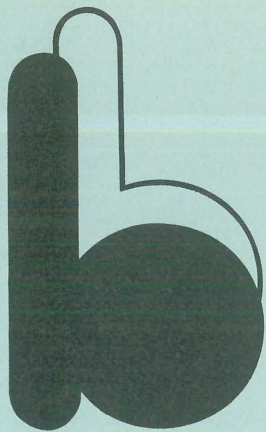
Accordo di fornitura tra Unicomp e Cromemco

È stato recentemente firmato a Mountain View (California) un accordo di fornitura tra la società Unicomp s.r.l. di Cinisello, importatrice in Italia di micro e minisistemi di calcolo, e la società californiana Cromemco.
L'accordo -che è stato firmato tra il Presidente della Cromemco, Dr. Harry Y. Garland e l'amministratore delegato della Unicomp, Ing. Pietro di Camillo - garantisce la fornitura per i prossimi anni di tutti i prodotti a catalogo della Cromemco e i relativi supporti per servire il mercato italiano.
La Cromemco, nata quattro anni fa con prodotti dedicati al settorre del personal computing, si è rapidamente conquistata una posizione predominante producendo sistemi basati sul microprocessore Zilog Z80 e capaci di gestire fino a sette terminali in linea: il Sistema Tre è stato recentemente ancor più potenziato con l'introduzione di floppy disks a doppia faccia e di unità a dischi che permettono di soddisfare le esigenze di calcolo e gestionali delle piccole e medie aziende.
La Unicomp s.r.l. è stata la prima società a riconoscere l'adattabilità al mercato italiano di questi nuovi tipi di prodotti, non solo nel settore hobbistico, ma soprattutto in quello della gestione aziendale.

Un elaboratore che pesa 350 grammi

È inglese il più piccolo personal computer del mondo.
È una creazione della Sinclair Research, una azienda pressoché sconosciuta se nella denominazione non comparisse il nome Sinclair. È stato infatti Clive Sinclair a progettare lo ZX80 (così si chiama il personal computer in questione), dopo essersi già distinto in passato in altri importanti sviluppi, come le prime calcolatrici elettroniche.
Lo ZX80, basato su un microprocessor Z80A della Nippon Eletric, ha le stesse dimensioni di un registratore portatile a cassette e pesa non più di 360 grammi. Il costo non arriva alle 200.000 lire, compreso il trasformatore di corrente. Ha bisogno, come supporto, di un televisore e presenta una tastiera sensoriale.
L'apparecchio è accompagnato da manuale che ne spiega il funzionamento anche ai non addetti ai lavori. Si tratta ora di vedere se il Clive riuscirà a gestire industrialmente la sua invenzione. Alla sua Sinclair Research sono già arrivate molte offerte: alcune sono proposte di collaborazione, altre richieste di acquistare la licenza di produzione.

I tre microprocessori dell'elaborazione personale

Si fanno grossi sforzi commerciali per diffondere i personal computers, ma l'aspetto forse ancora più importante di questi sistemi sono i microprocessori. Del ventaglio di dispositivi oggi disponibili tre appaiono i favoriti nella scelta da parte dei costruttori dei piccoli sistemi di elaborazione. Si tratta di tre microprocessori da 8-bits: l'8080 della Intel, lo Z80 della Zilog e il 6502 della MOS Technology e naturalmente le varie versioni costruite da seconde sorgenti. L'8080 è stato il microprocessor largamente usato nella fase di decollo dell'industria, funzione successivamente presa a carico dallo Z80 in considerazione della sua maggiore potenza. Un altro dispositivo-chiave si è rilevato il 6502 (realizzato da MOS Technology, Synertek e Rockwell) oggi impiegato come cuore di personal computers quali Commodore Business

bits & bytes

la prima azienda italiana specializzata in software per il TRS-80 Lev II

è lieta di annunciare l'apertura dei propri uffici in Milano e di presentare i programmi finora disponibili

B 001 - Mailing list — **L. 130.000**
Crea e gestisce un elenco illimitato di indirizzi. Sorting a 3 chiavi e 2 sottochiavi di ricerca più 3 opzioni aggiuntive. Visualizzazione su monitor e stampa completamente formattate. 4 opzioni di stampa con ricerca automatica del posizionamento.
16 K + LPrinter

B 002 - Aggiornamento mailing list — **L. 40.000**
Permette l'aggiornamento degli elenchi creati con il precedente programma. Eliminazione, correzione ed inserimento di indirizzi. Registrazione su nastro elenco aggiornato. Visualizzazione su monitor completamente formattata.
16 K

B 001 + B 002 Package mailing list ed aggiornamento — **L. 150.000**

B 010 - Word processing — **L. 180.000**
Consente la creazione e la gestione completa di un testo. Registrazione del testo su nastro, editing, posizionamento e marginatura automatici, numerose altre opzioni di stampa tra cui centratura simmetrica, stampa al vivo, incolonnamenti etc.
16 K + LPrinter

P 001 - Totocalcio - sistema a correzione di errori — **L. 60.000**
Impostando una qualsiasi probabile colonna vincente e le varianti di correzione (doppie, triple o miste), questo programma sviluppa tutte le possibili colonne contenenti il numero di errori desiderato. Calcolo preventivo del numero di colonne del sistema e del relativo costo.
16 K

P 002 - Totocalcio - chiave alfa 6 super — **L. 50.000**
Sistema di 10 triple con alto rapporto di riduzione (13,5). Scelte le 3 fisse, il programma elabora tutte le colonne del sistema, visualizzandole su monitor o stampandole su tabulati.
16 K + opz. LPrinter

S 001 - Compurhythm — **L. 75.000**
Questo programma elabora il bioritmo primario di una persona (individuale) o, simultaneamente, di una coppia (comparato). Creazione nastro nominativi. Opzione prelievo nominativi manuale da tastiera o automatico da nastro. Opzione visualizzazione su monitor o stampa, entrambe completamente formattate. Evidenziazione giorni apicali, periodici e mezzo-periodici. Stampa contestuale della data e del giorno della settimana. Opzione di stampa spiegazioni dettagliate con grafico. Analisi finale armonia di ritmo delle coppie.
16 K + opz. LPrinter

S 020 - Test della personalità — **L. 25.000**
Attraverso 70 domande, questo programma di psicoanalisi indaga sugli aspetti più salienti della vostra personalità, dandovene un quadro analitico conclusivo.
16 K

G 020 - Missione nello spazio — **L. 50.000**
Anno 3100: astronavi nemiche hanno invaso la galassia e minacciano la Terra. Siete inviato in missione con l'ordine di annientarle. Missili, campi magnetici, scudi protettivi, radar a corto e lungo raggio, computer di bordo sono alcuni tra i comandi di cui è dotata la vostra nave spaziale. Un programma assai complesso e dotato di grafici, che non mancherà di coinvolgervi!
16 K

G 001 - Trappola mortale — **L. 25.000**
Duello all'ultimo sangue contro robots muniti di sensori termici che vi rincorrono da un capo all'altro del monitor, cercando di uccidervi. Un programma grafico con scelta iniziale del livello di difficoltà.
16 K

G 010 - Chi è? — **L. 20.000**
Un simpatico gioco di conversazione con il computer, che vi fa delle domande e cerca di indovinare a chi o a che cosa state pensando. Possibilità di impostare la conversazione su persone, animali, oggetti, etc.
16 K

Tutti i nostri programmi, disponibili su cassette magnetiche di alta qualità, sono dotati di blocchi automatici che riducono al minimo l'eventualità di commettere errori. L'incisione viene eseguita su un solo lato del nastro ed almeno due volte, al fine di garantire la massima fedeltà di riproduzione.

Spese di spedizione e I.V.A. 14% incluse.

Per ordinazioni, vi preghiamo di compilare il tagliando a fianco riportato e di spedire a:

Bit

Nome ______________________

Via ______________________

Città ______________ CAP __________

Cod. Fiscale ______________________

Sigle dei programmi desiderati ______________

Allego ass. banc. N. ______________

Autorizzo contrassegno ☐

per l'importo totale di L. ______________

bits & bytes Viale Monte Grappa, 4 - 20124 Milano - Tel. 02-651628

Comparazioni logiche per localizzazione errori.

Analisi in tempo reale per ricerca guasti.

Manutenzione, test o ricerca, lo strumento è uno solo.

L'esigenza di analizzare la funzionalità logica di dispositivi elettronici non cessa allorquando i «digits» sono, per così dire, usciti dalle mani dei progettisti. Come per i volt e per gli ampère, la «logica» segue il prodotto attraverso lo sviluppo ed il collaudo e via via durante la manutenzione ed il «service»: aree ben al di fuori della fabbrica, ove la ricerca di eventuali anomalie di funzionamento della logica dei circuiti hanno sempre costituito un arduo problema.
Così è stato sin ora ed è perciò che abbiamo progettato un nuovo strumento, il **Logicoscopio PM3540** destinato a soddisfare tutte queste esigenze dal laboratorio al servizio esterno di manutenzione.
Il PM3540 è il solo strumento singolo portatile in grado di fornire sia la rappresentazione delle tabelle di dati sia la loro rappresentazione in funzione del tempo. Premendo un pulsante il PM3540 **si trasforma da logic analiyzer in un oscilloscopio** doppia traccia, con triggering eccitato dalla stessa parola logica, e con **esatto** inizio dell'oscillogramma al bit iniziale della parola di trigger (requisito essenziale per ricerca guasti digitali). Si possono: visualizzare i dati in codice binario, ottale o esadecimale: catturare blocchi da 64 x 16 bits ovunque lungo tutto il flusso dei dati e scorrerli pagina dopo pagina col solo tocco di un pulsante; scoprire la posizione ed analizzare le cause di eventuali errori; utilizzare lo strumento come un normale oscilloscopio.
La versatilità del PM3540, equipaggiato con una dotazione di sonde logiche dedicate, può essere evidenziata da una lunga serie di significanti prerogative in aggiunta ad un'eccellente funzionalità quale oscilloscopio a doppia traccia.
Letteralmente meglio delle **due funzioni analisi in tempo reale ed analisi degli stati logici** con un solo strumento, compatto, portatile e di alta classe.

Logicoscopio PM3540.

PHILIPS

Philips S.p.A. - Divisione Scienza & Industria
V.le Elvezia, 2 - 20052 Monza - Tel. (039) 36.35.248
Filiali: BOLOGNA (051) 493.046
CAGLIARI (070) 666.740 PADOVA (049) 657.700
ROMA (06) 342.041 TORINO (011) 2.164.121

Machines e Apple. Una buona penetrazione sta avendo il 6800 della Motorola. Ci sono poi degli outsiders. Alcuni fabbricanti di sistemi per l'elaborazione personale, cioè, hanno fatto uso di microprocessori un po' particolari in questo settore. È il caso del CDP 1802 della RCA (usato dalla stessa casa per i sistemi VIP) e del CP-1610 a 16 bits della General Investment (usato dalla Mattel Electronics per la serie Intellec).

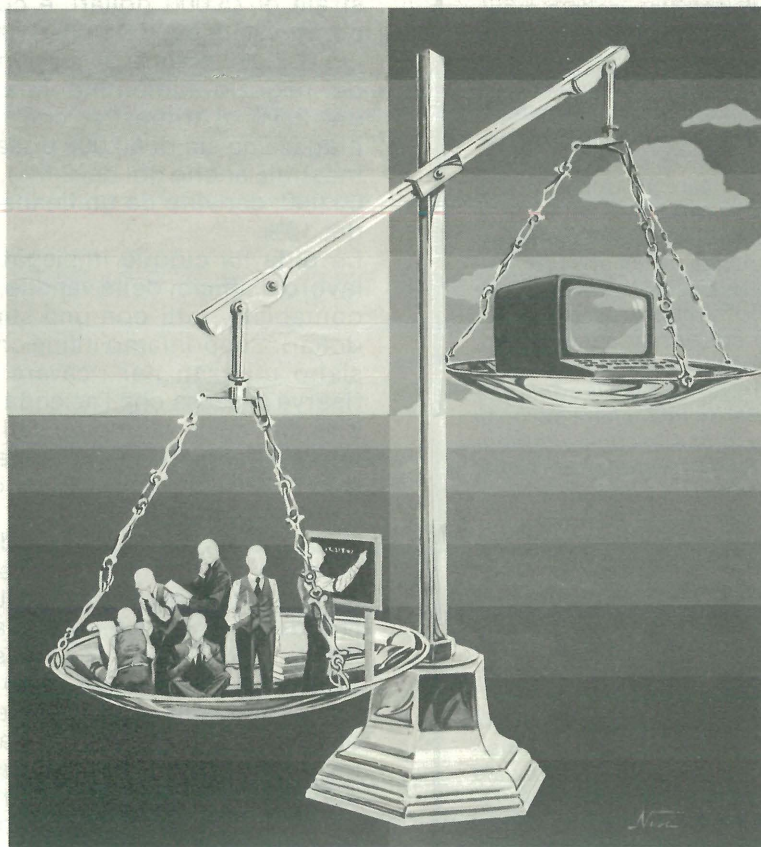
L'IBM lavora in gran segreto alla tecnologia Ga As

I problemi che in passato avevano frenato, soprattutto a livello di fabbricazione, l'impiego dell'arseniuro di gallio nella realizzazione di chips ultrarapidi sembrano prossimi ad essere risolti in modo soddisfacente. Diverse società sono tornate a credere nell'arseniuro di gallio, e di conseguenza hanno ripreso ad investire in maniera adeguata. Fra queste ci sono la Hewlett-Packard (avrebbe già raggiunto un livello di 150 gates integrati su un elemento), la Rockwell e la IBM. Con la tecnologia Ga As quest'ultima prevede di potere addirittura far meglio che con il processo conosciuto come Josephson Junction, con il quale sta progettando un calcolatore da 70 milioni di operazioni al secondo, che dovrebbe essere pronto nel 1984. Rispetto a questa tecnologia, che necessita di molto freddo per operare, la tecnologia Ga As ha il grosso vantaggio di poter operare a temperature normali.

Per le famiglie MCS-80 ed 85 L'Intel annuncia il Pascal

L'Intel aggiunge il Pascal-80 ai linguaggi ad alto livello già disponibili (PL/M, BASIC, FORTRAN, e ICIS — COBOL).
Il Pascal-80 è disponibile su dischetto a singola o doppia densità e gira sotto il sistema operativo ISIS-II sui modelli INTELLEC Serie II e MDS 800. Tale linguaggio offre, rispetto al Pascal originale del prof. Wirth, delle estensioni che lo rendono più flessibile e adatto alle applicazioni gestionali e di controllo industriale. Il Pascal-80 si compone di un compilatore e di un sistema run-time.
Per creare e modificare i programmi-sorgente scritti in Pascal, si possono utilizzare i mezzi di sviluppo standard offerti dalla Intel, come per esempio il CREDIT: il compilatore accetta in ingresso il programma — sorgente e lo converte in un codice intermedio noto come P-Code. Il sistema run-time esegue questo codice intermedio in modo interpretativo sotto ISIS-II.
A proposito della Intel c'è poi una grossa notizia. La società starebbe lavorando ad un microprocessore con un set di istruzioni analogo a quello di un IBM 370. Ciò significherebbe offrire un calcolatore da tavolo con 128 Kbytes di memoria principale e 20 megabytes su disco ad un costo di 2000 dollari circa.

Accordo PBS-ALF

È stata recentemente costituita a Bari la ALF Software & Sistemi S.r.l.
La ALF ha siglato un accordo con la PBS di Milano, distributrice per l'Italia del microcomputer MISTRAL 801, per la commercializzazione nell'Italia meridionale di questa macchina. La ALF, inoltre, curerà lo sviluppo di software applicativo per vari campi di attività.
Al momento attuale sono disponibili diversi packages applicativi che riguardano i seguenti campi: Ingegneria Civile Strutture, Ingegneria Termotecnica, Gestione piccole imprese, Word processing & lettere commerciali, Ingresso dati.
In particolare per quanto concerne il package di ingegneria civile sono stati sviluppati i seguenti programmi: Trave continua ad n campate, Trave di fondazione rigida su suolo elastico, Trave di fondazione elastica su suolo elastico alla Winkler, Analisi dei carichi per edifici multipiani, Calcolo agli stati limite per le strutture in c.a. (Sollecitazione di sforzo normale, Sollecitazione di flessione, Sollecitazione di sforzo normale e flessione, Progetto e verifica allo sforzo di taglio). Progetto o verifica a flessione col metodo N.
Questi programmi potranno essere acquistati sia in blocco, sia singolarmente, rivolgendosi alla ALF - Via Capruzzi 192, BARI.

Considerazioni sui microcomputers per applicazioni gestionali: i costi aziendali

Parte II di **R. Williams**

È questo il secondo di una serie di quattro articoli sulla implementazione del calcolatore nelle piccole imprese, rivolta ai responsabili di questo tipo di azienda, che spesso non hanno familiarità con la problematica connessa con i calcolatori. Scopo di questi articoli è fornire delle indicazioni che permettono di valutare obbiettivamente quando la computerizzazione è una scelta economica in rapporto alle proprie esigenze, e, di conseguenza, quando effettuarla.
Le Parti I e II affrontano il problema di *quando* acquistare un calcolatore, la III e la IV di *come* farlo. Nella parte I (Bit n. 6) è stato sviluppato il concetto di analisi differenziale, all'interno di un discorso generale su costi, vantaggi economici e fattibilità dell'acquisizione di un calcolatore (v. il riassunto che daremo più avanti). La Parte II si serve di questi concetti per analizzare un'azienda specifica. La Parte III tratterà della scelta e dell'ottimizzazione dell'hardware e del software; la Parte IV infine illustrerà come di fatto si articola l'acquisizione di un calcolatore, sotto il duplice aspetto delle prestazioni ottenibili e degli accorgimenti da usare.

RIASSUNTO DELLA PARTE I
Considerazioni economiche generali

L'aspetto più importante da ricordare della Parte I è il concetto di analisi differenziale, mediante il quale le differenze nel movimento di cassa risultanti dall'acquisizione di un calcolatore possono essere illustrate in rapporto alla situazione di non computerizzazione. Dato le definizioni di entrata, uscita e movimento di cassa differenziali, e di accumulo di cassa differenziale (ossia il movimento di cassa differenziale rapportato ad un determinato intervallo di tempo), si è visto che inizialmente per alcuni trimestri, si ha un accumulo di cassa differenziale tendente progressivamente a valori negativi, in corrispondenza all'accumularsi dei costi del calcolatore senza una compensazione in termini di vantaggi economici; in un secondo momento, realizzandosi questi vantaggi, l'accumulo di cassa differenziale cessa di decrescere, ed assume l'andamento opposto, fino a raggiungere il punto zero, o di ammortamento, per poi continuare a salire rapidamente col passar del tempo.
In questa analisi spiccano alcuni aspetti particolarmente interessanti: il più sorprendente è la constatazione che un qualsivoglia ritardo implicherà dei costi inevitabili e permanenti, che indicano come altamente rischiosa la soluzione di NON computerizzazione; un altro è che il calcolatore vi è visto come un investimento di sicura convenienza, nel senso che si ammortizza da solo. L'unico aspetto problematico è quello della fattibilità, cioè se l'azienda è in grado di far fronte ai primi trimestri di accumulo di cassa differenziale negativo. Se la computerizzazione non è fattibile, le perdite differenziali dovute alla non computerizzazione sono una disgraziata conseguenza della sottocapitalizzazione.
La parte I fornisce anche un'analisi particolareggiata delle condizioni di fattibilità più favorevoli, consistenti in un giusto equilibrio fra entrata differenziale ed uscita differenziale. Infatti sono date delle valutazioni sui costi ottimali dell'hardware (7.000 ÷ 10.000 dollari), sui costi dei packages di software (sui 2.000 dollari) e sui costi della programmazione di personalizzazione (attorno ai 600 dollari mensili per un anno). Per quanto concerne i vantaggi, si presume un risparmio sul costo del lavoro di 1/2 - 5 impiegati, e un incremento dei profitti lordi del 2 - 10$.

Introduzione alla Parte II

I concetti, le prospettive, i costi ed i vantaggi presentati nella Parte I possono essere applicati ad una generica piccola azienda. In questa Parte II ne diamo un esempio.
Nel primo paragrafo si definiscono i dati dell'azienda ipotetica, quali il campo di attività ed i parametri finanziari e fiscali, oltre ad una serie di costi e di vantaggi determinati dal calcolatore, il tutto espresso in cifre.
Nel secondo paragrafo si applica l'analisi differenziale a tutti questi dati, con l'aiuto di tabelle, al fine di valutare gli effetti economici sull'azienda in termini di denaro e di valore netto.
Nell'ultimo paragrafo si esamina come applicare l'analisi ad un'azienda non più ipotetica, ma reale. Vi si parla dei costi finanziari, dei limiti del credito, della remunerazione dell'investimento e dell'utilizzo del capitale, non trascurando il ruolo che hanno l'intuito e il buon senso.

Caratteristiche dell'azienda ipotetica

Definiamo la nostra azienda ipotetica come un negozio di vendita al dettaglio, con un ricavo lordo trimestrale di 75.000 dollari, e costi trimestrali, stipendi compresi, di 69.750 dollari: la differenza attiva è perciò di 5.250 dollari. Si supponga che i ricavi ed i costi del negozio aumentino, in assenza del calcolatore, del 2,5% al trimestre; che il valore delle merci in magazzino sia di 60.000 dollari (con un aumento trimestrale anche qui del 2,5%), e che 20.000 dollari, sui 60.000, derivino da un finanziamento al tasso annuo del 15%.
La ditta ha cinque impiegati, che si occupano del lavoro d'ufficio, delle vendite, delle spedizioni e della contabilità, tutti con uno stipendio mensile di 600 dollari. Supponiamo infine che i fondi d'investimento siano utilizzati per ricavare un interesse su quelle riserve di cassa che l'azienda non può in realtà assorbire, per cui al momento dell'installazione del calcolatore si abbia un accantonamento di 6.000 dollari, una riserva di cassa attiva di 1.000 dollari ed una riserva del credito nulla.
I costi d'implementazione del calcolatore saranno analoghi a quelli definiti nella Parte I, eccetto che per una più accentuata preoccupazione di economia: l'hardware costerà 6.000 dollari, il software 1.000 dollari, la programmazione presso il cliente sarà limitata a 3 mesi-uomo, per un costo globale di 4.500 dollari. L'acquisto dell'hardware e del software, per il valore di 7.000 dollari, avverrà nella forma del *leasing*, con un acconto del 10% e dei pagamenti mensili di 175 dollari per la durata di tre anni: allo scadere del terzo anno, la ditta potrà optare per l'acquisto, versando 3.500 dollari. La programmazione presso il cliente avverrà a metà tempo per un periodo di sei mesi.
Anche i vantaggi economici previsti dall'acquisizione del calcolatore saranno analoghi a quelli visti nella nella Parte I, con la differenza che saranno significativi gradualmente, con un andamento lineare, per tutto l'intervallo di un anno, successivo alla conclusione della programmazione di personalizzazione. In particolare, si presume che il calcolatore porterà ad un risparmio di lavoro nella fatturazione, nella tenuta dei libri contabili, e nella gestione del magazzino, nella misura di un impiegato. Si prevede anche che la migliorata gestione del magazzino, resa più efficiente dall'analisi delle vendite passate e presenti, comporterà un incremento annuale del rapporto di approvigionamento del 20%.
Anche la gestione delle vendite migliorerà, grazie all'analisi del personale addetto e dell'efficienza della pubblicità, congiunta a programmi motivazionali ed a sondaggi continui della clientela: questo comporterà, senza un aumento delle spese di gestione, un incremento del 5% dei ricavi lordi in più rispetto al normale sviluppo della ditta senza il calcolatore.
Inoltre la migliorata gestione del lavoro aumenterà l'efficienza dell'esercizio, o ridurrà le spese di gestione. Questo potrà essere raggiunto tramite: l'eliminazione di lavori non essenziali, l'ottenimento degli elenchi dei termini e dei pagamenti, l'impiego di più

raffinate strategie finanziarie rese possibili dall'analisi dell'attività e delle potenzialità dell'azienda. In conclusione supponiamo che le spese generali diminuiranno del 3% in rapporto alla situazione di non computerizzazione, e senza che questo comporti variazioni nelle vendite.
Questi costi e vantaggi rispecchiano la situazione tipica che l'imprenditore medio potrà trovarsi di fronte, e che si aspetterà, in seguito all'acquisizione di un calcolatore.
Il passo successivo consiste nell'analisi di questi dati approssimativi al fine di valutare l'impatto economico sull'azienda e quindi determinare la fattibilità dell'acquisto del calcolatore.

Analisi dell'azienda

Secondo quanto si è detto nella Parte I, calcoliamo l'entrata, l'uscita, il movimento di cassa e l'accumulo di cassa differenziali. Questi calcoli sono supportati dalle Tabelle I — VII.

Entrata differenziale

Le voci più importanti dei ricavi differenziali, come si è già detto, sono date dall'accresciuta efficienza delle vendite, dal risparmio di lavoro, dall'accresciuta efficienza dell'amministrazione e da una migliore gestione del magazzino, e sono rappresentate rispettivamente nelle Tabelle I, II, III e IV. In tutte queste tabelle si assume la soluzione *senza calcolatore* come punto di riferimento in base al quale si valuta la soluzione con calcolatore; il tempo è specificato mediante il numero di trimestre Tn, per cui n individua i dati presenti alla fine del trimestre Tn, essendo l'installazione del calcolatore avvenuta a T0.
La tabella I (tabella dell'efficienza delle vendite) definisce in primo luogo i parametri dell'impresa relativi alla soluzione *senza calcolatore*. Come si è già detto in precedenza, i ricavi hanno una crescita del 2,5% al trimestre, il profitto netto rappresenta il 7% del ricavo lordo (5.250/75.000), e di conseguenza i costi rappresentano il 93% del ricavo. Inoltre, dato che il costo della merce rappresenta il 60% del valore del negozio, il costo del venduto rappresenta il 60% del ricavo: pertanto le spese di gestione sono date dalla differenza fra le spese ed il costo del venduto.
La soluzione *con calcolatore* è caratterizzata da un ricavo identico a quello che si ha nella soluzione *senza calcolatore*, fino a quando la programmazione non è completa: via via che il calcolatore entra in un ritmo regolare di attività, l'efficienza delle vendite sale linearmente, trimestre per trimestre. La differenza nel ricavo, rispetto alla situazione *senza calcolatore* comincia a delinearsi al termine del terzo trimestre (a T3), in ragione dell'1, 25%, e si stabilizza al valore massimo del 5% al termine del sesto trimestre (a T6).
Dato che il costo del venduto è uguale al 60% del ricavo, le spese di gestione si mantengono identiche a quelle della situazione *senza calcolatore*. Allora i costi sono dati dalla somma delle spese di gestione più il costo del venduto, ed in base a questo si calcola l'utile relativo alla soluzione *con calcolatore* il ricavo differenziale (mostrato nell'ultima colonna della Tabella I) consiste nel conseguente accrescimento dell'utile rispetto alla soluzione *senza calcolatore*.
il risparmio sui costi del lavoro è illustrato nella Tabella II: con uno stipendio di 600 dollari al mese per cinque impiegati, nella situazione *senza calcolatore* si hanno dei costi trimestrali di 9.000 dollari. La soluzione *con calcolatore* non prevede risparmi sul costo del lavoro se non dopo la fase di programmazione presso il cliente, analogamente a quanto si è visto nella precedente analisi; allo stesso modo, ancora, il risparmio ha un andamento lineare nell'anno successivo, partendo da un minimo di 1/4 di impiegato a T3 ad un massimo di 1 impiegato a T6. Il ricavo differenziale è dato dalla differenza nei costi del lavoro per ciascun trimestre.
La tabella III illustra gli effetti del calcolatore sull'efficienza della gestione: pur essendo i ricavi gli stessi, alla fine le spese di gestione diminuiranno del 3% in conseguenza dell'impiego del calcolatore. In questa

TABELLA I - RICAVI DIFFERENZIALI
EFFICIENZA DELLE VENDITE
(incremento del 5% nelle vendite a parità di spese di gestione)

DATA (TRIM.)	COMMENTI	SENZA calcolatore					CON calcolatore				ricavo differenziale (profitto)
		ricavi	spese	costo del venduto	spese di gestione	profitti	ricavi	costo del venduto	spese	profitto	
0	Install. calcolatore	75000	69750	45000	24750	5250	75000	45000	69750	5250	0
1	Programm.-Vantaggi zero	76875	71494	46125	25369	5381	76875	46125	71494	5381	0
2	↓	78797	73281	47278	26003	5516	78797	47278	73281	5516	0
3	Vantaggi=1,25% delle vendite	80767	75113	48460	26653	5654	81777	49066	75719	6058	404
4	Vantaggi=2,5 % delle vendite	82786	76991	49672	27319	5795	84856	50914	78233	6623	828
5	Vantaggi=3,75% delle vendite	84856	78916	50913	28003	5940	88038	52823	80826	7212	1272
6	Vantaggi=5,0 % delle vendite	86977	80887	52186	28701	6090	91326	54796	83497	7829	1739
7		89151	82911	53491	29420	6240	93609	56166	85586	8023	1783
8		91380	84984	54828	30156	6396	95949	57570	87726	8223	1827
9		93665	87108	56199	30909	6556	98348	59009	89918	8430	1874
10		96006	89286	57604	31682	6720	100807	60485	92167	8640	1920
11		98406	91518	59044	32474	6888	103327	61997	94471	8856	1968
12		100867	93806	60520	33286	7061	105910	63547	96833	9077	2016
13		103388	96151	62033	34118	7237	108558	65135	99253	9305	2068
14		105973	98555	63584	34971	7418	111272	66764	101735	9537	2119
15		108622	101019	65173	35846	7603	114054	68433	104279	9775	2172
16	↓	111338	103544	66803	36741	7794	116905	70144	106885	10020	2226

TABELLA II - RICAVI DIFFERENZIALI				
RISPARMIO LAVORO (1 impiegato)				
DATA (TRIM.)	COMMENTI	SENZA calcolatore	CON calcolatore	ricavo diff. (costo del lavoro)
		costo del lavoro	costo del lavoro	
0	Install. calcolatore	9000	9000	0
1	Programm.-Vantaggi zero	9000	9000	0
2	↓	9000	9000	0
3	Vantaggi=1/4 di persona	9000	8550	450
4	Vantaggi=1/2 di persona	9000	8100	900
5	Vantaggi=3/4 di persona	9000	7650	1350
6	Vantaggi=1 persona	9000	7200	1800
7		9000	7200	1800
8		9000	7200	1800
9		9000	7200	1800
10		9000	7200	1800
11		9000	7200	1800
12	↓	9000	7200	1800

valutazione naturalmente il ricavo, il costo del venduto, i costi complessivi e le spese di gestione relativi alla situazione *senza calcolatore* sono gli stessi che compaiono nella Tabella I. La soluzione *con calcolatore* evidenzia la diminuzione lineare delle spese di gestione, che comincia a delinearsi a T3 con il valore dello 0,75%, per raggiungere il 3% a T6. Il ricavo differenziale è dato quindi dal ridursi delle spese di gestione in seguito alla computerizzazione.
L'ultimo ricavo differenziale preso in considerazione è quello che risulta dalla migliorata gestione del magazzino (v. Tabella IV). A T0 l'inventario ha il valore di 60.000 dollari, e, nella situazione *senza calcolatore*, questa cifra sale, in accordo con il crescere dell'azienda del 2,5% al trimestre. Definito il *rapporto di approvvigionamento* come costo del venduto diviso per il valore dell'inventario, nella situazione *con calcolatore* il rapporto di approvvigionamento nell'inventario aumenta con andamento lineare, da un minimo del 5% a T3 ad un massimo del 20% a T6. Dato che il valore dell'inventario si calcola con la formula

$$(\text{inventario}) = \frac{(\text{costo del venduto})}{(\text{rapporto di approvvigionamento})}$$

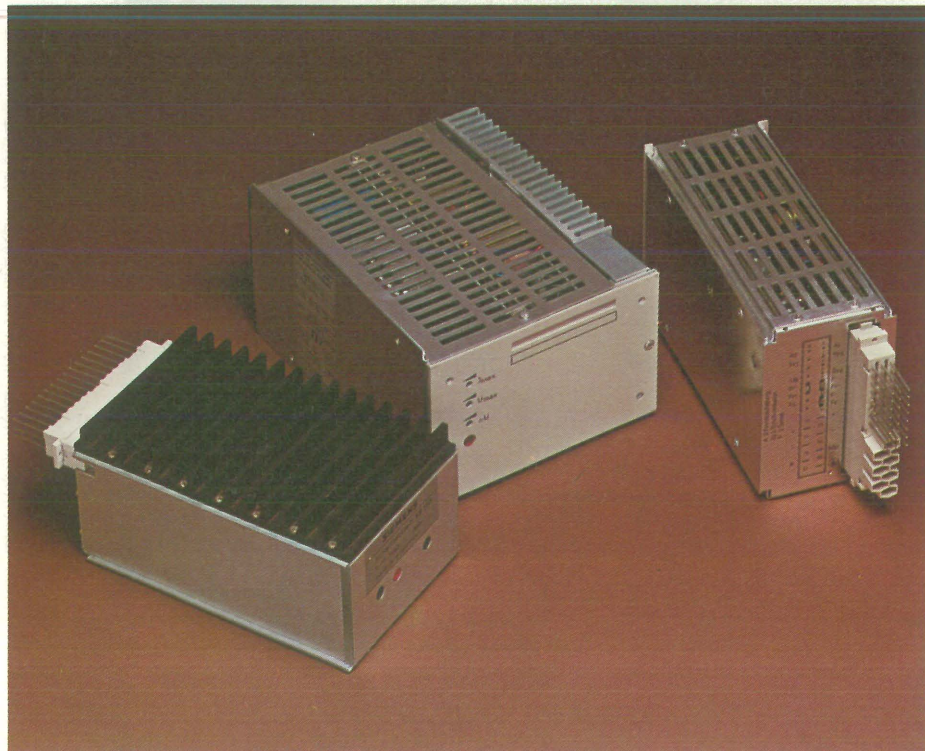
TABELLA III - RICAVI DIFFERENZIALI							
EFFICIENZA GESTIONE diminuz. del 3% delle spese di gestione a parità di ricavi							
DATA (TRIM.)	COMMENTI	ricavi	costo del venduto	SENZA calcolatore		CON calcolatore	ricavo differ. (spese di gestione)
				spese	spese di gestione	spese di gestione	
0	Install. calcolatore	75000	45000	69750	24750	24750	0
1	Programm.-Vantaggi zero	76875	46125	71494	25369	25369	0
2	↓	78797	47278	74281	26003	26003	0
3	Vantaggi 0,75% spese gestione	80767	48460	75113	26653	26453	200
4	Vantaggi 1,5 % spese gestione	82786	49672	76991	27319	26909	410
5	Vantaggi 2,25% spese gestione	84856	50913	78916	28003	27373	630
6	Vantaggi 3 % spese gestione	86977	52186	80887	28701	27840	861
7		89151	53491	82911	29420	28537	883
8		91380	54828	84984	30156	29250	906
9		93665	56199	87108	30909	29982	927
10		96006	57604	89286	31682	30731	951
12		98406	59044	91518	32474	31500	974
13		100867	60520	93806	33286	32287	999
14		103388	62033	96151	34118	33094	1024
15		105973	63584	98555	34971	33922	1049
16		108622	65173	101019	35846	34770	1076
17	↓	111338	66803	103544	36741	35639	1102

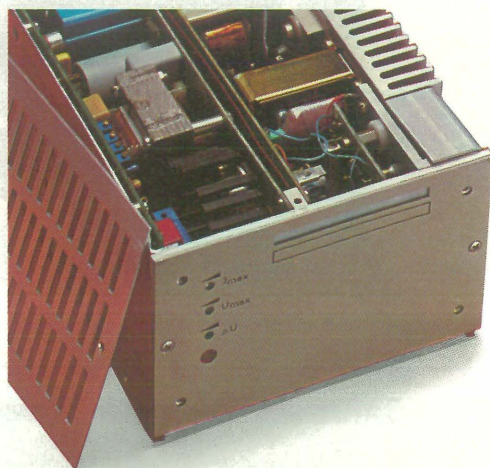
TABELLA IV - RICAVI DIFFERENZIALI						
GESTIONE MAGAZZINO Increm. del 20% del rapp. di approvv.						
DATA (TRIM.)	COMMENTI	SENZA calcolatore	CON calcolatore			ricavo differ. (interesse)
		valore magazzino	valore magazzino	fondo pareggio magazzino	interesse	
0	Install. calcolatore	60000	60000	0	0	0
1	Programm.-Vantaggi zero	61500	61500	0	0	0
2	↓	63038	63038	0	0	0
3	Vantaggi=5 % rapp. approvv.	64613	61536	3077	0	0
4	Vantaggi=10% rapp. approvv.	66229	60208	6021	115	115
5	Vantaggi=15% rapp. approvv.	67884	59030	8854	226	226
6	Vantaggi=20% rapp. approvv.	69582	57985	11597	332	332
7		71321	59434	11887	435	435
8		73104	60920	12184	446	446
9		74932	62443	12489	457	457
10		76805	64004	12801	468	468
11		78725	65604	13121	480	480
12		80693	67244	13449	492	492
13		82711	68925	13786	504	504
14		84778	70648	14130	517	517
15		86898	72415	14483	530	530
16	↓	89070	74225	14845	543	543

il contante reso disponibile da questo processo è poi investito in un fondo di pareggio dell'inventario, che rende un tasso d'interesse annuo del 15%: questi interessi rappresentano il ricavo differenziale, perché non esisterebbero senza un miglioramento della gestione del magazzino.
Ai fini di una maggior chiarezza della Tabella IV, si tenga presente che l'incasso degli interessi del fondo di pareggio dell'inventario è registrato non nello stesso trimestre in cui si è effettuato il deposito, ma in quello successivo.

Uscita differenziale

Nell'impresa ipotetica che stiamo esaminando le uscite differenziali sono limitate ad un solo tipo, e cioè il costo diretto dell'acquisizione del calcolatore. Questo è illustrato nella Tabella V, che tiene conto sia dall'hardware che del software.
In questa tabella sono stati introdotti i segni meno e più, in base alla convenzione, da considerarsi valida d'ora in poi, secondo la quale il segno più sta a indicare un movimento di denaro che va all'esterno verso l'azienda, mentre il segno meno designa il movimento inverso. Alla soluzione *senza calcolatore* corrispondono molto semplicemente valori nulli, a sottolineare il fatto che questa soluzione è presa come punto di riferimento. Per la soluzione *con calcolatore* si parte dal presupposto di una combinazione affitto-acquisto per l'hardware ed il software (in packages), con un anticipo di 700 dollari, ed un canone di affitto mensile di 175 dollari. Allo scadere del terzo anno si esercita l'opzione di acquisto versando in più 3.500 dollari, per cui questa somma, più l'acconto iniziale, (in totale 4.200 dollari) rappresenta la svalutazione del calcolatore dopo tre anni. La programmazione di personalizzazione viene pagata, per un periodo di sei mesi e per prestazioni a mezza giornata, 750 dollari al mese. (La quota mensile di 600 dollari di cui si è detto nella Parte I si riferiva ad un contratto più lungo).
La combinazione di acquisto illustrata è comunque solo una delle molte possibilità esistenti (quali ad

gli alimentatori "switched mode" serie VB di ridotto hanno solo l'ingombro e il peso

L'ottimizzazione di alimentatori a "chopper", così come il rispetto delle prescrizioni relative all'eliminazione dei disturbi, richiedono una vasta esperienza.

Ed è proprio dall'esperienza conseguita dalla Siemens nella progettazione e produzione di apparecchiature per telecomunicazioni e telex, oltre che di impianti di misura, comando e regolazione, che sono nati gli alimentatori "switched mode" della serie VB.

tipo	tensione di ingresso	tensione di uscita	potenza
VB 103 N VB 103 ND	220V~	5V, 5V uscite 12V, 12V isolate	75...95W
VB 106 N	220V~	5V, 5V uscite 15V, 15V isolate	75...95W
VB 116/5 VB 116/12 VB 116/15 VB 116/24	220V~ 220V~ 220V~ 220V~	+ 5V +12V +15V +24V	60W 60W 60W 60W
VB 119	220V~	+5,1V, +30V −5V, +12V −12V non regolata	90W (picco 150W)
VB 150/5 VB 150/12 VB 150/15 VB 150/24	220V~ 220V~ 220V~ 220V~	+5V +12V +15V +24V	125W 125W 125W 125W
VB 401/5 VB 401/15 VB 401/15N VB 401/24	24V– / 25V~ 32V– / 25V~ 32V– / 25V~ 48V– / 38V~	+5V +12...+15V regolabile −12...−15V +24V	50W 150W 150W 100W

Elevato rendimento = ingombro e peso ridotti

L'elevato rendimento (75÷85% a seconda dei tipi) che caratterizza gli alimentatori della serie VB ha reso possibile una notevole riduzione del peso e delle dimensioni degli alimentatori stessi.
Essi rappresentano quindi la risposta ottimale a tutte le problematiche di installazione in cui i fattori di ingombro e peso risultano determinanti.

Per qualsiasi informazione, vi preghiamo di rivolgervi direttamente alla Siemens Elettra S.p.A., 20124 Milano, Via Fabio Filzi 25/A, Tel. (02) 6248
Divisione sistemi e componenti elettronici - Reparto A 203.

Siemens. Per fare dei piccoli alimentatori ci vuole una grande esperienza

TABELLA V - USCITA DIFFERENZIALE

COSTO CALCOLATORE E PROGRAMMAZIONE
leasing α 175/mese e programm. per 6 mesi a metà tempo

DATA (TRIM.)	COMMENTI	costi SENZA calco-	CON calcolatore: costi: hardware	software	impos. sul patrimonio	TOTALE	correzioni sui costi: detraz. fiscale sulle spese	credito di imposta sull'investim.	crediti di ammortam.	uscita differ. NETTA
0	Install. Calcolatore									
	Versamento acconto	0	—700	0	0	—700	0	0	0	—700
1	Canone affitto +									
	Programm. metà tempo	0	—525	—2250	0	—2775	+694	0	0	—2081
2		0	—525	—2250	0	—2775	+694	0	0	—2081
3		0	—525	0	0	—525	+131	0	0	—394
4		0	—525	0	0	—525	+131	0	0	—394
5	Canoni d'affitto	0	—525	0	0	—525	+131	0	0	—394
6		0	—525	0	0	—525	+131	0	0	—394
7		0	—525	0	0	—525	+131	0	0	—394
8		0	—525	0	0	—525	+131	0	0	—394
9		0	—525	0	0	—525	+131	0	0	—394
10		0	—525	0	0	—525	+131	0	0	—394
11		0	—525	0	0	—525	+131	0	0	—394
12		0	—525	0	0	—525	+131	0	0	—394
13	Opzione acquisto + manutenz.	0	—3575	0	—32	—3607	+27	+70	+53	—3457
14	Manutenzione	0	—75	0	—32	—107	+27	+70	+53	+43
15		0	—75	0	—32	—107	+27	+70	+53	+43
16		0	—75	0	—32	—107	+27	+70	+53	+43

esempio l'acquisto immediato, e l'acquisto finanziato, l'affitto senza possibilità di acquisto, e l'affitto con possibilità di acquisto, ma in tempi diversi). Per quanto riguarda la programmazione di personalizzazione, si possono avere altre soluzioni, che prevedono quantità totali diverse di lavoro di programmazione e di addestramento. La scelta fra queste soluzioni è condizionata da fattori quali la fattibilità dal punto di vista economico, l'esigenza di un accordo il più armonico possibile con il sistema di contabilità sia a breve che a lungo termine, senza contare il peso di altri fattori di natura più soggettiva.

Il movimento di cassa differenziale e l'accumulo di cassa differenziale

I calcoli fatti di entrata e di uscita differenziali sono riassunti schematicamente nella Tabella VI.
Si può notare che il movimento di cassa differenziale e l'accumulo di cassa differenziale hanno andamenti simili a quelli presentati, in termini generali, nella Parte I, con la differenza che ora si riferiscono ad un'impresa specifica con proiezioni reali, che hanno un ammortamento più rapido.
Le entrate differenziali che compaiono nella Tabella VI sono tratte dalle Tabelle I - IV. La somma di queste quantità è ridotta di una quota d'imposta del 25%, per confermarla alle uscite differenziali desunte dalla Tabella V, sulla quale si è ipotizzata un'analoga riduzione.
Il movimento di cassa differenziale è la somma dell'entrata e dell'uscita differenziale, calcolata tenendo presenti i segni, secondo la convenzione stabilita.

TABELLA VI - MOVIMENTO DI CASSA DIFFERENZIALE E ACCUMULO DI CASSA DIFFERENZIALE

DATA (TRIM)	COMMENTI	ENTRATA DIFFERENZIALE: effic. delle vendite	risparmio lavoro	efficienza gestione	gestione magazzino	TOTALE (meno 25% tasse)	uscita differenz. TOTALE	movimento di cassa differ.	accumulo di cassa differ.	interesse fondo di pareggio trim. precedente tasse incluse	accumulo di cassa differ. (interessi compresi)	accumulo di cassa differ. ritardato di 2 trim.	costo del ritardo
0	Install calcolatore	0	0	0	0	0	—700	—700	—700	0	—700	0	+700
1	Affitto+programm. metà tempo												
	Vantaggi zero	0	0	0	0	0	—2081	—2081	—2781	—26	—2807	0	+2807
2		0	0	0	0	0	—2081	—2081	—4862	—104	—4992	—700	+4292
3	Affitto+vantaggi	404	450	200	0	792	—394	+398	—4465	—182	—4778	—2807	+1971
4		828	900	410	115	1689	—394	+1295	—3170	—167	—3650	—4992	—1342
5		1272	1350	630	226	2609	—394	+2215	—955	—119	—1554	—4778	—3224
6		1739	1800	861	332	3549	—394	+3155	+2200	—36	—1565	—3650	—5215
7		1783	1800	883	425	3676	—394	+3282	+5482	+83	+4930	—1554	—6484
8		1827	1800	906	446	3734	—394	+3340	+8822	+206	+8476	+1565	—6911
9		1874	1800	927	457	3794	—394	+3400	+12222	+331	+12207	+4930	—7277
10		1920	1800	951	468	3854	—394	+3460	+15682	+458	+16125	+8476	—7649
11		1968	1800	974	480	3917	—394	+3523	+19205	+588	+20236	+12207	—8029
12		2016	1800	999	492	3980	—394	+3586	+22791	+720	+24542	+16125	—8417
13	Acquisto+manutenz.	2068	1800	1024	504	5047	—3457	+590	+23381	+855	+25987	+20236	—5751
	e vantaggi												
14	Manutenz.+vantaggi	2119	1800	1049	517	4114	+43	+4157	+27538	+877	+31021	+24542	—6479
15		2172	1800	1076	530	4184	+43	+4227	+31765	+1033	+36281	+25987	—10294
16		2226	1800	1102	543	4253	+43	+4296	+36061	+1191	+41768	+31021	—10747

L'accumulo di cassa differenziale è un totale cumulativo del movimento di cassa. Si assume che questo accumulo di cassa venga investito in un fondo di pareggio generale che produce un interesse del 20% (che è un interesse più alto di quello prodotto dal fondo di pareggio dell'inventario, a causa delle meno impellenti esigenze di liquidità). Per ciascun trimestre, l'interesse proveniente dal fondo di pareggio viene calcolato con riferimento, al trimestre precedente, e ridotto quindi della quota d'imposta del 25%: il valore così ottenuto è quello che compare nella tabella, e inoltre viene utilizzato per creare una versione corretta dell'accumulo di cassa differenziale.
Il costo del ritardo si calcola confrontando l'accumulo di cassa differenziale con l'identica quantità ritardata di due trimestri. Nel confronto, in cifre, il segno più sta ad indicare un attivo di cassa determinato dal ritardo nella computerizzazione, il segno meno indica un passivo di cassa.
Nella tabella VII sono illustrati gli utili netti, pagate le tasse. Gli utili relativi alla soluzione, *senza calcolatore* sono desunti dalla Tabella I, e quindi ridotti della quota d'imposta del 25% per conformarli alla situazione della Tabella VI. Per la soluzione *con calcolatore* si procede sommando il movimento di cassa differenziale, l'interesse prodotto dal fondo di pareggio ed i profitti ottenuti *senza calcolatore*: nella nostra impresa con l'impiego del calcolatore gli utili netti sono quasi raddoppiati. (Si noti che l'interesse prodotto dal fondo di pareggio non è stato incluso nel movimento di cassa differenziale, perchè il suo calcolo utilizza proprio il movimento di cassa differenziale).
La tecnica del fondo di pareggio rappresenta un'estensione dell'analisi differenziale, tesa a tener conto del pareggio di cassa in modo diverso da un semplice contobanca, in quanto comporta il concetto di manipolazioni attive sulla cassa. L'accumulo di cassa differenziale è necessariamente assorbito o creato dal pareggio di cassa dell'azienda, a seconda del segno. D'altra parte, questo fondo può assumere la forma di svariate combinazioni di finanziamento, investimenti in azioni e titoli, restituzione o impiego di pretesti contratti dalla ditta, o infine investimento nella ditta stessa.

TABELLA VII - PROFITTI

DATA (TRIM.)	PROFITTI NETTI PAGATE LE TASSE (25%)		variazione percentuale profitti netti
	SENZA calcolatore	CON calcolatore (Interessi compresi)	
0	3938	3238	—18%
1	4036	1929	—52%
2	4137	1952	—53%
3	4240	4456	+ 5%
4	4346	5474	+ 26%
5	4455	6551	+ 47%
6	4566	7685	+ 68%
7	4680	8045	+ 72%
8	3797	8343	+ 74%
9	4917	8648	+ 76%
10	5040	8958	+ 78%
11	5166	9277	+ 80%
12	5295	9601	+ 81%
13	5428	6873	+ 27%
14	5564	10598	+ 90%
15	5703	10963	+ 92%
16	5845	11332	+ 94%

Considerazioni per un'azienda specifica

L'analisi che abbiamo svolto è un esempio di metodo di calcolo negli effetti economici complessivi, nell'ipotesi che tutti i costi ed i vantaggi previsti siano noti in precedenza. La determinazione di questi fattori è un fatto estremamente specifico, perchè dipende dalle caratteristiche particolari dell'azienda, dalla sua organizzazione e dalle eventuali carenze.
I fattori di costo si possono ragionevolmente circoscrivere entro il prezzo totale dell'hardware e del software, come è stato mostrato nella Parte I, ma esistono, e se ne è già parlato, molte tecniche di acquisizione di un calcolatore, ciascuna delle quali presenta un impatto diverso sulla situazione finanziaria dell'azienda, e può quindi richiedere un'analisi differenziale preliminare sulla cui base valutare e scegliere la soluzione più adatta. Ad esempio, quest'analisi differenziale preliminare potrebbe trascurare l'entrata differenziale, che resta invariata, poiché non cambiano le prestazioni del calcolatore.
Fra i molti fattori da tener presenti in un'analisi differenziale ci sono costi finanziari, la remunerazione dell'investimento, i limiti del credito, l'impiego di capitale, le imposte varie, ed altri fattori ancora. Così ad esempio il pagamento in contanti può essere abbastanza costoso se l'impresa è al limite del credito, ed il capitale richiesto remunera in modo costante l'investimento. Alternativamente, un'azienda può sostenere l'acquisto ad un costo netto inferiore a quello di un acquisto diretto, allorquando il funzionamento rende disponibile del denaro liquido che possa venir investito con degli utili superiori al tasso d'interesse del finanziamento.
Le considerazioni di ordine fiscale rivestono un'importanza particolare. Ad esempio, l'acquisto beneficia dei crediti fiscali sull'investimento e dei crediti di ammortamento, ma è penalizzato dalle imposte sul patrimonio, dai costi reali dell'ammortamento e dal costo del capitale. Le soluzioni di leasing sono spesso molto più convenienti dal punto di vista economico a causa del limitato costo iniziale del capitale impiegato e degli effetti minimi sui limiti del credito, ma d'altra parte il leasing comporta concretamente tassi d'interesse più elevati, una non disponibilità dei crediti fiscali sull'investimento e l'assenza dei crediti di ammortamento.
In casi particolari, come nella programmazione di personalizzazione, l'acquisto immediato del programma comporta un elevato costo iniziale del capitale, controbilanciato però dalla messa in attività immediata del calcolatore, con i vantaggi economici che ne conseguono. Invece le soluzioni che comportano tempi più lunghi riducono sì il fabbisogno iniziale di capitale, ma portano un ritardo nell'entrata differenziale, con la possibile conseguenza di un movimento di cassa netto più gravoso.
Valutare quali sono gli elementi intervenendo sui quali è possibile ottenere benefici è più difficile. Ad esempio, l'azienda è ben strutturata e funziona dal punto di vista organizzativo, oppure si hanno spre-

chi, e lavoro inutili? Si fa fronte ai pagamenti nei tempi prescritti? Vengono ignorati i clienti potenziali per mancanza di note e sondaggi? Capita che buona parte del personale sia inattiva in certi periodi di tempo o sia occupata in lavori non prioritari, sopratutto a scapito di altri più importanti? Qualè la mole di lavoro nella contabilità, e quanto è completamente utile, e quanto parzialmente superfluo? In quale percentuale il magazzino è costituito da fondi? Quali articoli si vendono bene? Si perde tempo, e clienti, perchè non si conosce la situazione degli ordini arretrati? Quante entrate potenziali vanno perse per un'analisi finanziaria insufficiente?
Queste domande sono un esempio di come si debba procedere per definire, in termini quantitativi, i vantaggi economici che ci si devono aspettare dall'impiego di un calcolatore. Esse sono molto legate alle situazioni specifiche, e la valutazione richiede una piena conoscenza dell'azienda, sia sul piano dell'intuito che su quello oggettivo. Il segreto, come si è visto nell'analisi che abbiamo svolto, sta nello scomporre i vantaggi economici previsti in effetti ridotti, ben delimitati e distinti, e poi combinare questi effetti in un'analisi differenziale della cassa.

Sommario

Abbiamo definito un'impresa ipotetica, stabilito i costi ed i guadagni previsti in seguito alla computerizzazione, e quindi calcolato gli effetti economici della computerizzazione sotto forma di movimento di cassa differenziale. Ma oltre a questo, abbiamo sottolineato l'importanza dell'intuito e della creatività, requisiti indispensabili ad un imprenditore per poter definire e valutare uno per uno tutti i fattori che intervengono nella determinazione dei costi e dei guadagni legati alla computerizzazione in modo consono alla singola impresa specifica. L'analisi differenziale è *soltanto* un mezzo esatto per valutare degli effetti, e si limita a prendere in esame le operazioni effettuate dall'imprenditore. L'analisi valuta le scelte, ma è l'imprenditore a determinare le scelte da valutare. ■

TRIBUNA

L'HP-85 e l'evoluzione del personal computer

di Alice Pantera - Hewlett-Packard Italiana

I personal computer, sistemi di calcolo accessibili quasi a tutti per prezzo, dimensioni e facilità d'uso, hanno subito negli ultimi tempi una notevole diffusione.

Il motivo della loro larga espansione sul mercato risiede principalmente nelle caratteristiche dei computer stessi. Innovazioni tecnologiche come il microprocessore, il linguaggio di programmazione evoluto e le unità periferiche miniaturizzate che si possono collegare a questi sistemi, ne hanno consentito una distribuzione su larga scala.

Anche la Hewlett-Packard ha realizzato l'importanza di questo settore e pur se con un certo ritardo, si è affacciata in questo mercato con un prodotto veramente professionale, l'HP-85, un personal computer che si viene a collocare in una posizione intermedia fra i calcolatori tascabili professionali e i computer da tavolo colmando uno spazio fino ad oggi non ancora occupato da altri costruttori.

La Hewlett-Packard infatti si era preposta di realizzare un prodotto a misura di professionista e quindi nella progettazione ha tenuto conto delle esigenze di questa categoria che necessita di una configurazione minima iniziale, di un prodotto veramente portatile e soprattutto compatto. È nato così l'HP-85 l'unico "personal" sul mercato che si presenti come un sistema integrato in una singola unità che comprende la CPU (16K), la tastiera, lo schermo, (che anche se piccolo è ad alta risoluzione) la stampante e la memoria di massa.

A questa configurazione di partenza, che è il punto di arrivo frutto di diversi aggiornamenti per molti altri costruttori, è possibile collegare altre periferiche alcune delle quali espressamente prodotte per l'HP-85 in modo da formare un sistema completo.

Anche le politiche d'introduzione dell'HP-85 riflettono lo stesso orientamento che ne ha influenzato la progettazione: così prima è stato introdotto il calcolatore, poi verranno le periferiche in modo da affrontare il mercato passo a passo seguendo lo stesso andamento del professionista che prima acquista il calcolatore, impara ad utilizzarlo al meglio e poi identifica le proprie esigenze di espandibilità aumentando la sua potenza di calcolo attraverso le periferiche sempre più sofisticate che gli verranno offerte dall'HP.

A questo proposito vale la pena di sottolineare che l'evoluzione tecnologica dell'HP tiene sempre conto di chi ha già investito nel prodotto base.

In particolare, essendo l'HP-85 il capostipite di una famiglia destinata a durare a lungo nel tempo, il suo acquirente effettua un investimento finalizzato a medio e lungo termine.

I mercati verso i quali l'HP punta con questo "personal" sono sostanzialmente due: il mercato scientifico (tradizionale per l'HP) e il mercato commerciale. Per questa ragione la distribuzione dell'HP-85 è stata affidata ad una struttura di vendita suddivisa in due gruppi:
uno che si occupa della vendita indiretta attraverso una limitata rete di rivenditori e l'altro che si occupa del mercato tecnico (hardware OEM) e dei clienti industriali.

Ne consegue che l'utente privato si potrà avvalere anche dell'assistenza del rivenditore che ha preventivamente seguito i corsi di addestramento in HP.

Anche l'assistenza tecnica, che per ora è gestita completamente dall'HP, verrà in futuro quasi certamente delegata ai rivenditori, una volta che il loro personale avrà completato la propria preparazione tecnica in HP.

Il software dell'HP-85 è un altro punto di vantaggio che l'HP offre ai rivenditori. Infatti mentre l'HP si preoccupa della preparazione di packages a larga diffusione di uso generale, i rivenditori/consulenti (o software-house) potranno dedicarsi alla produzione di packages applicativi personalizzati per il mercato verticale.

Inoltre l'introduzione delle nuove periferiche e l'ampia possibilità di interfacciamento anche a periferiche non Hewlett-Packard, farà sicuramente nascere la possibilità di sviluppare, a livello di rivenditore stesso, piccoli sistemi commerciali venduti chiavi in mano.

Attualmente la rete di rivenditori dell'HP-85 è abbastanza esigua, ma questa è stata una scelta motivata alla limitata disponibilità del prodotto. Secondo le tradizionali politiche di produzione Hewlett-packard, un nuovo prodotto non deve rivoluzionare l'assetto di una divisione, in questo caso la Personal Computing Products, ma deve porsi come prodotto aggiuntivo che contribuisce al profitto della società, senza recare detrimento ai prodotti di successo già esistenti e dei quali viene continuata la produzione.

Tuttavia entro l'anno, non appena la capacità produttiva della divisione comincerà ad assumere livelli tali da soddisfare maggiormente le richieste del mercato, altri punti di vendita qualificati verranno ad aggiungersi a quelli già esistenti.

SALOTA

MULTI FUNCTION COMPUTER "SALOTA MCF 512" DESCRIZIONE HARDWARE

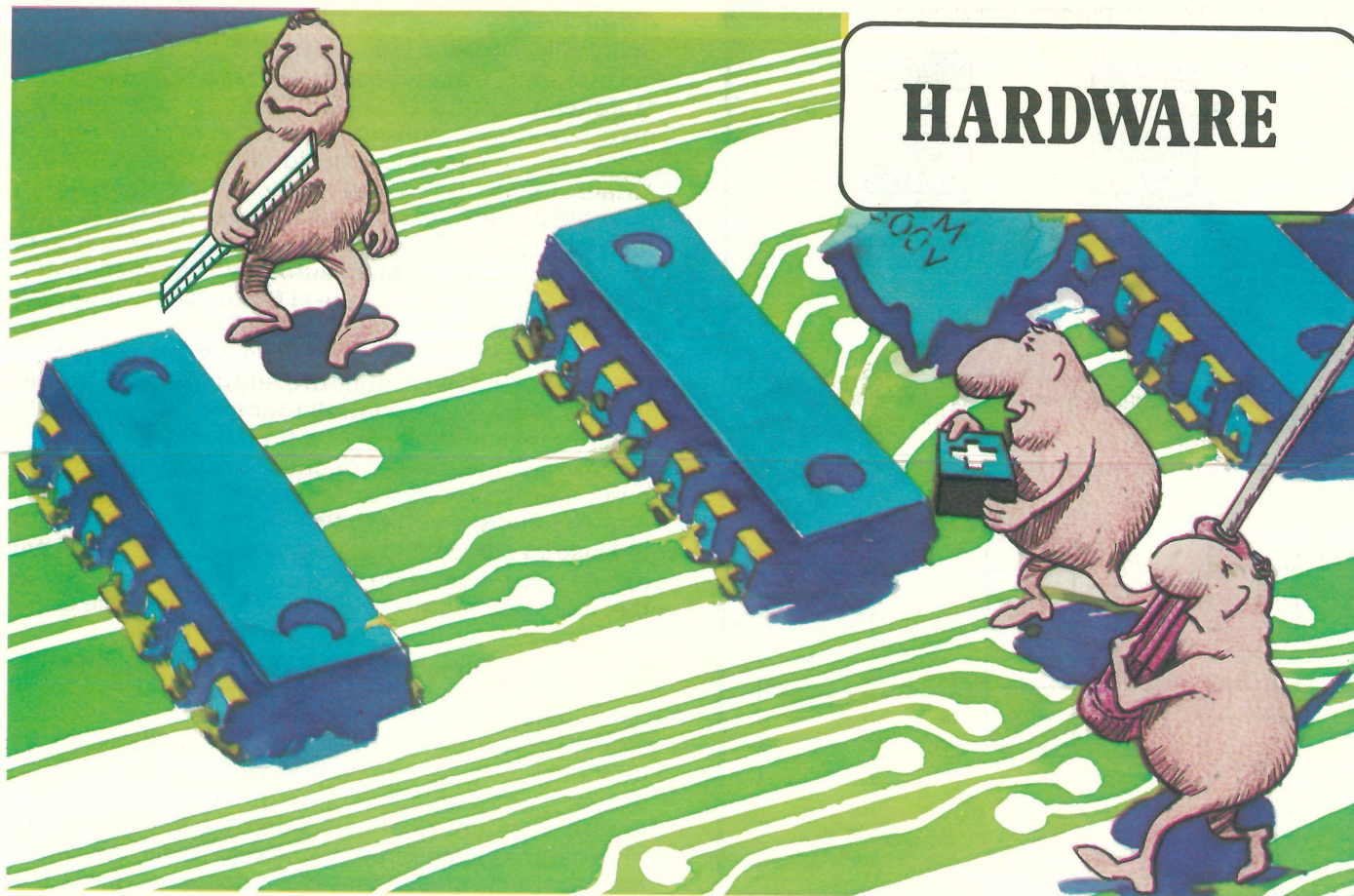
UNITÀ CENTRALE

RAM:	da 16 KB a 512 KB in banchi da 16 KB oppure 64 KB.
ROM:	1 KB residente per caricamento D.O.S.
I/O:	2 INTERFACCE SERIALI (RS 232 C / V 24) 2 INTERFACCE PARALLELE 1 CURRENT LOOP 20mA FINO AD UN MASSIMO DI 7 GRUPPI
FLOPPY CONTROLLER:	permette la gestione da 1 a 4 FLOPPY DISK DRIVER SINGOLA O DOPPIA FACCIA.
FLOPPY DISK DRIVER:	STANDARD BASF 6104 (Doppia faccia) capacità unitaria 512 KB. op BASF 6102 (Singola faccia) Capacità unitaria 256 KB. PERSCI 277 (Doppio floppy driver).
HARD DISK CONTROLLER:	codifica FMD per dischi C.D.C.
HARD DISCK:	dischi fissi 8" da 24 MB dischi fissi in tecnologia WINCHESTER da 64 MB a 256 MB. dischi mobili da 16 MB.
TRASMISSIONE DATI:	metodo FULL DUPLEX HALF DUPLEX tipo SINCRONA ASINCRONA
BAUD RATE:	da 110 Baud a 76800 Baud.
MICROPROCESSORE:	ZILOG Z 80 A (4 MHz)

PERIFERICHE

STAMPANTI:	a matrice da 80 colonne 80 cps a 132 colonne 600 cps WORD PROCESSING da 10 cps a 680 cps 132 colonne.
TERMINALI:	video con tastiera 1920 chr 12 inch 1920 chr 16 inch tastiera e video SEPARABILI.
PLOTTER:	size 15" (X) 10" (Y) 22" (X) 17" (Y) compatibili RS 232 C ASCII
SOFTWARE DI BASE:	CDOS SISTEMA OPERATIVO (CP/M COMPATIBILE) CBASIC - COBOL - FORTRANT IV - PASCAL MACRO ASSEMBLER - WORD PROCESSING DATA BASE - BASIC MULTI-USER

Via Curtatone, 16 — 20098 S. Giuliano Milanese (MI) — Tel. 02-9880147

HARDWARE

PICOCOMPUTER: Interfacciamento con registratore a cassette

Parte V di **D. Del Corso**

Chiunque si sia consumato le dita su una qualsivoglia tastiera sa che il primo indispensabile complemento di ogni sistema a microprocessore, anche di dimensioni minime come il Picocomputer, è una memoria non volatile da utilizzare come archivio per i diversi programmi da caricare nella RAM.

Obiettivi e specifiche di progetto

Prestazioni e prezzi delle memorie di massa variano su diversi ordini di grandezza; la soluzione presentata in queste pagine come prima estensione del Picocomputer ha come principale obiettivo il basso costo e prevede l'uso di un normale registratore a cassette. Questa scelta va a scapito essenzialmente del tempo di accesso, in quanto i programmi vanno cercati "a mano" sulla cassetta, e della velocità di trasferimento, che è limitata dalle caratteristiche del canale.
Le principali caratteristiche del sistema di memoria di massa qui presentato per il Picocomputer sono quindi:

— uso di un normale registratore a cassette di media qualità;

a cura del Gruppo Amatoriale Micro-Elaboratori, Torino *e del Laboratorio Didattico di Elettronica Applicata dell'Istituto di Elettronica e Telecomunicazioni del Politecnico di Torino.*

— semplici circuiti per il modulatore e demodulatore;
— comando manuale del registratore;
— controllo di tutte le operazioni tramite il solo Picoperiferico.

In un sistema per registrazione e rilettura su cassetta possiamo individuare i blocchi funzionali di *Figura 1*. Con questa organizzazione ciascun modulo opera ad un ben definito livello ed è relativamente indipendente dagli altri. Ad esempio è possibile cambiare tipo di modulazione agendo solo sui blocchi C ed E, o modificare la struttura dei messaggi tramite i moduli A e G. Come per ogni progetto che impiega il microprocessore, esistono diverse alternative per la ripartizione dei compiti tra hardware e software; le funzioni dei blocchi B ed F possono essere svolte da una interfaccia seriale tipo UART o USART, o dal programma, e lo stesso modulatore/demodulatore può essere realizzato almeno in parte a software. Nel nostro caso si è scelto di implementare a programma quante più funzioni possibile; solamente i blocchi C ed E sono realizzati con interfacce specifiche, per sfruttare al massimo le caratteristiche del canale.

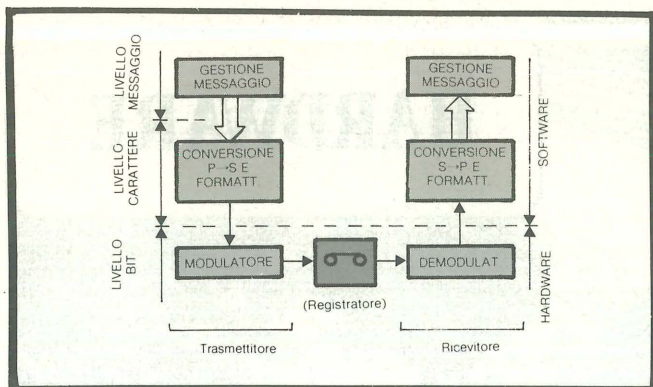



Figura 1 - Organizzazione di un sistema di memoria di massa a cassette.

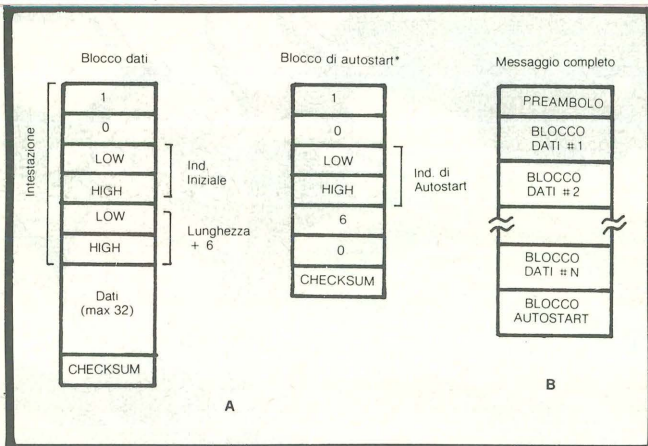



Figura 2 - Formato dei blocchi e dei messaggi.

Formato dei messaggi

I dati immagazzinati sulle cassette devono essere organizzati in messaggi che contengono tutte le informazioni necessarie per la fase di rilettura, cioé indirizzo a cui vanno scritti i dati, lunghezza del messaggio, controlli di errore.

Il formato scelto prevede l'invio diretto del contenuto delle celle di memoria su caratteri di 8 bit. La struttura dei messaggi è analoga a quella della banda perforata dei calcolatori PDP 11 DEC. Rispetto ad altri formati che inviano due caratteri ASCII per ogni byte di memoria (ad esempio il cosiddetto standard "industriale" Intel), si ha il vantaggio di dimezzare, a pari quantità di informazioni, la lunghezza del messaggio. Il formato prevede blocchi dati per il trasferimento di informazioni e blocchi di "autostart" che specificano ove riprendere l'esecuzione al termine della lettura di un messaggio *(v. Figura 2a)*.

In chiusura dev'essere sempre inviato un messaggio di autostart. La rivelazione di eventuali errori è affidata ad una parola di controllo (checksum), calcolata come somma modulo 2 bit a bit di tutti i caratteri del messaggio. Il carattere di checksum è trasmesso in coda a ciascun blocco. Per ridurre la probabilità di non segnalare errori multipli, la massima lunghezza di ciascun blocco è limitata a 32 bytes; messaggi più lunghi sono spezzati in una serie di blocchi consecutivi. Ogni messaggio *(v. Figura 2b)* comprende quindi:

— un preambolo, il cui scopo è chiarito in seguito;
— uno o più blocchi dati;
— un blocco di autostart.

Questo formato non prevede un'etichetta (cioé un nome) per i messaggi allo scopo di identificare i diversi programmi registrati, e quindi non è possibile in fase di lettura ricaricare solo un ben determinato programma, selezionato in base al nome, tra i molti registrati sulla stessa cassetta. Occorre portare il nastro in prossimità dell'inizio della registrazione che interessa (aiutandosi con il contatore e con commenti registrati a voce), e far partire la lettura da quel punto.

Per svincolarsi dalle variazioni di velocità del registratore, senza dover ricorrere a complicati circuiti per la separazione dati/portante, ogni carattere viene inviato in modo asincrono, completato da un bit di start e due bit di stop.

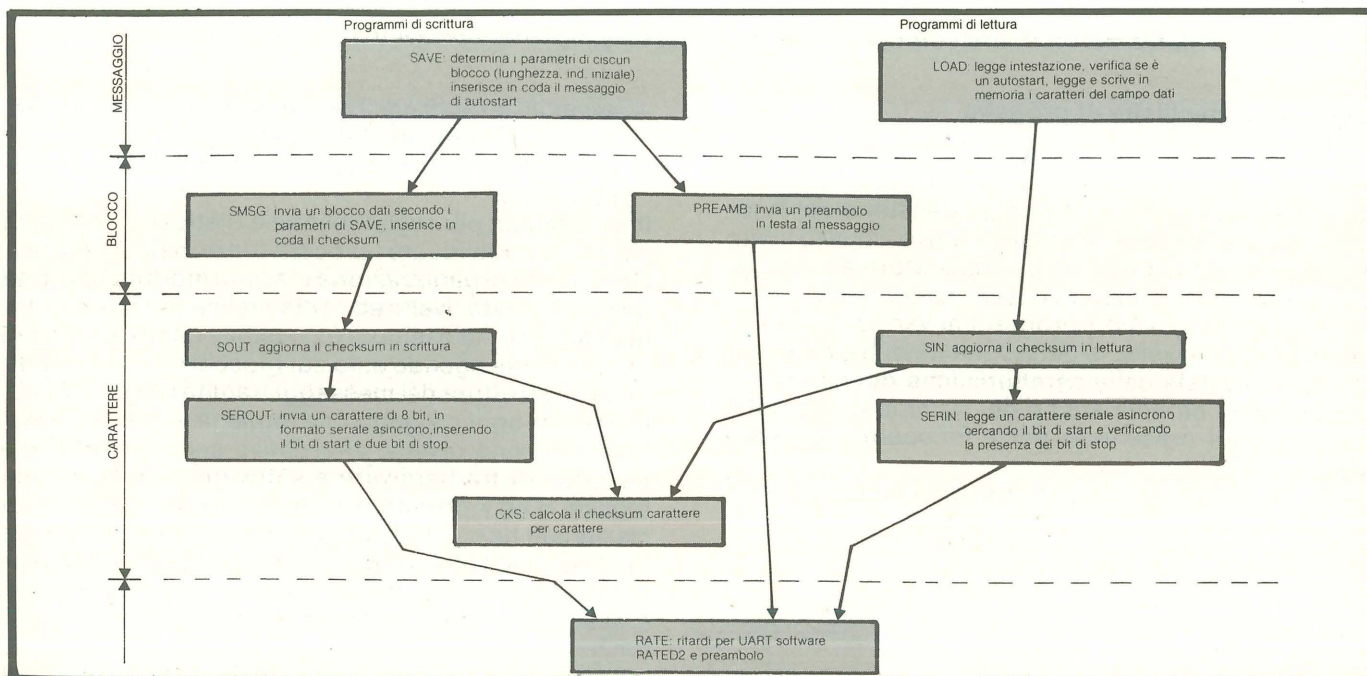



Figura 3 - Organizzazione del sofware di gestione.

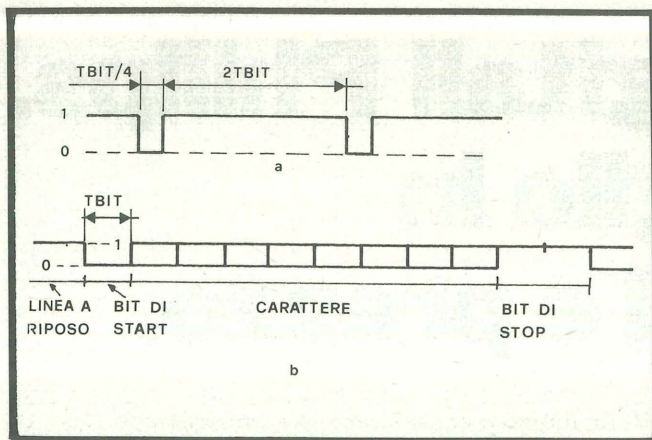



Figura 4 - Formato del preambolo (a) o di un carattere (b).

Programmi di gestione

Il complesso di funzioni da espletare per la gestione dei messaggi è suddiviso tra diversi sottoprogrammi organizzati in modo gerarchico, secondo lo schema di *Figura 3.* Ogni modulo viene chiamato da quelli di livello superiore ed usa a sua volta i sottoprogrammi di livello più basso.

SERIN e SEROUT costituiscono una vera e propria UART software; la cadenza di bit è specificata dal parametro DELY.

Prima di ogni registrazione viene inviato un preambolo, costituito da una serie di brevi pacchetti della portante. Scopo del preambolo è far intervenire il controllo automatico di guadagno del registratore per lavorare poi a livello costante nella successiva fase dati. Durante la lettura il preambolo viene completamente ignorato dalla UART software. Questo perché lo zero ha durata pari 1/4 T_{BIT}, e quindi è sempre interpretato come disturbo e non come bit di start di un carattere asincrono *(v. Figura 4).*

I diagrammi di flusso dei moduli software sono riportati nelle *Figure 5 e 6.*

I moduli principali SAVE e LOAD possono essere chiamati da monitor con un GOTO o inseriti come comandi direttamente nel monitor stesso.

Prima di attivare SAVE occorre predisporre i parametri (indirizzo iniziale e finale, autostart) nelle celle di memoria FROM, TO, ABUFF (vedi definizioni in testa al listato). SAVE termina con un RET, e può quindi essere chiamato come subroutine da programma di utente; lo stesso vale evidentemente per SMSG, SERIN, SEROUT, CKS.

Il modulo di lettura LOAD, una volta attivato, termina solo per un blocco di autostart o per un errore. Ogni carattere letto sul nastro viene inviato sulla porta CONADD e fa quindi comparire un carattere sul display; questo permette di seguire il corretto svolgimento delle operazioni. Durante l'operazione di LOAD vengono eseguiti due tipi di controllo di errore: a livello di singolo carattere si verifica la presenza di almeno un bit di stop, ed al livello di blocco viene controllato il checksum. In caso di errore termina l'operazione di lettura e viene segnalato, sul display del Picoperiferico, il codice di errore (sottoprogramma ERR). Viene considerata errore e trattata come tale anche una intestazione diversa da quella standard (1, 0). Questa uscita da LOAD può però essere usata anche per inserire blocchi dati particolari (ad esempio per l'identificazione dei programmi), individuati da una diversa intestazione. Dalla condizione di errore si esce solo con un RESET.

Figura 5 - Moduli per la scrittura dei messaggi.

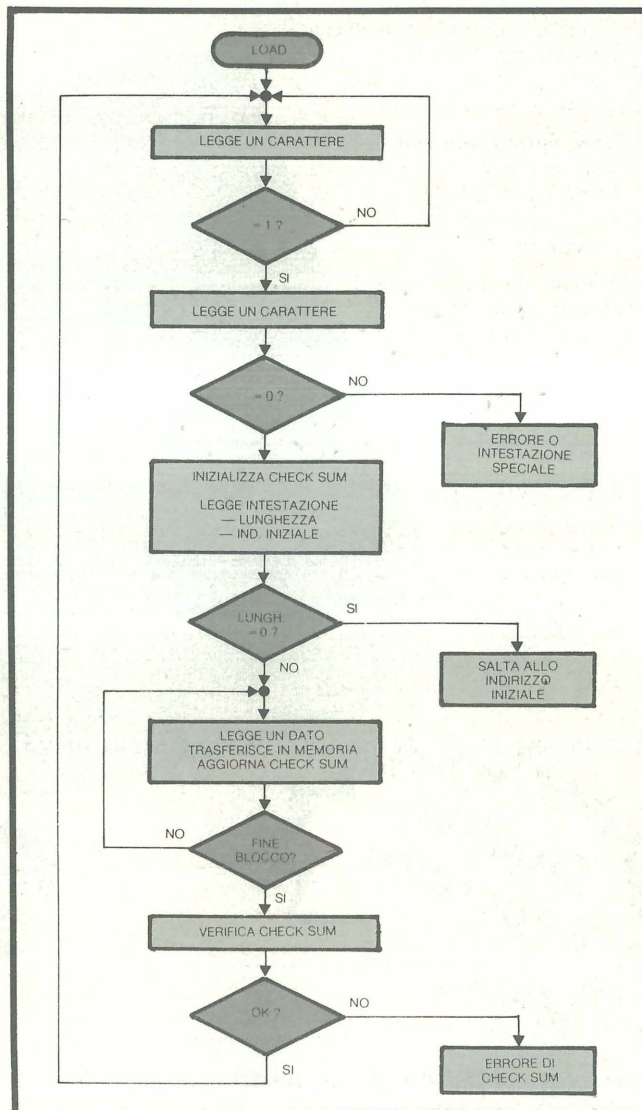



Figura 6 - Modulo per la lettura dei messaggi.

Multimetri digitali Philips. Il meglio in prestazioni e prezzo.

Da una analisi comparativa del rapporto prestazioni/prezzo i Multimetri Digitali **PM 2517** risultano vincenti.
Pur fornendo superbe prestazioni da strumenti di laboratorio quali le quattro cifre piene e le gamme automatiche, vengono offerti ad un prezzo altamente competitivo.

Vi invitiamo a considerare le caratteristiche professionali sotto elencate, unitamente alla possibilità di scegliere tra il modello con display a cristalli liquidi e quello a LED, la realizzazione ergonomica, robusta e compatta e giudicare quindi la fondatezza della nostra asserzione.

Displays a 4 cifre piene: aumentata risoluzione rispetto ai 3 1/2 cifre. Inoltre indicatore dell'unità di misura.

Scelta tra LED e LCD: scegliete secondo le vostre preferenze.

Cambio gamma automatico: per praticità di misura. Naturalmente vi è anche quello manuale.

Vero valore efficace: il solo modo per misurare correttamente segnali in c.a. non perfettamente sinusoidali.

Elevata risoluzione ed accuratezza: grazie alle 4 cifre piene e l'elevata sensibilità.

Correnti sino a 10 A: la tendenza di utilizzare tensioni sempre più basse richiede tassativamente di poter misurare sino a 10 A.

Protezione dai sovraccarichi: è impossibile danneggiarlo.

Piccolo ma robusto: non fragile plastica o deboli commutatori.

Design ergonomico: funziona in ogni posizione, automaticamente

Misura anche le temperature: la sonda opzionale consente questa misura utilissima per la ricerca guasti.

Congelamento della misura indicata: un grande vantaggio ottenibile con lo speciale puntale opzionale.

Rispetta le norme internazionali: quali? Virtualmente tutte.

Qualità Superiore

Il multimetro a 4 cifre senza compromessi

Filiali: BOLOGNA (051) 712.897
CAGLIARI (070) 860.761/2/3
PADOVA (049) 657.700
PALERMO (091) 400.066
ROMA (06) 382.041
TORINO (011) 210.404/8

Philips S.p.A.
Sezione Scienza & Industria
Viale Elvezia, 2 - 20052 Monza
Tel. (039) 36.35.1

PHILIPS Test & Measuring Instruments

PHILIPS

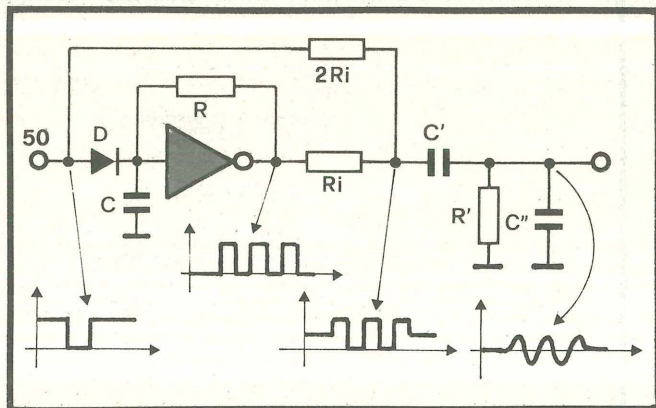



Figura 7 - Schema di principio del modulatore e relative forme d'onda.

Il listato completo, assemblato a partire da 100H per poter essere accodato a PICOMON.O, è CASS O in *Figura 11* (attenzione: lo stack pointer deve essere inizializzato dal programma chiamante, e per SAVE occorre predisporre un indirizzo di ritorno).

Modulatore/demodulatore

Questo modulo è necessario per adattare la banda del segnale alla banda del canale (registratore), e concorre a determinare la massima densità di registrazione (bit/sec o bit/cm) utilizzabile.
Una rassegna delle diverse tecniche di registrazione su supporto magnetico con indicazioni sulle prestazioni è contenuta nell'articolo *Codici per la registrazione magnetica digitale...*, di Graglia e Osella (Elettronica Oggi, n. 11, Novembre 1979).
Nel nostro caso, come compromesso tra semplicità, affidabilità e densità di registrazione, si è scelta la modulazione di ampiezza ON/OFF di una portante (AM al 100%). La densità è limitata principalmente dalle caratteristiche del registratore. Con il modulatore qui descritto si può arrivare a 1200 bit/sec; limitandosi a 600 bit/sec si ha una elevata affidabilità, ed il segnale può essere inviato su qualunque canale audio (anche su linea telefonica, attraverso accoppiatori acustici), con ottimi risultati.
Mantenendo gli stessi programmi di gestione, con

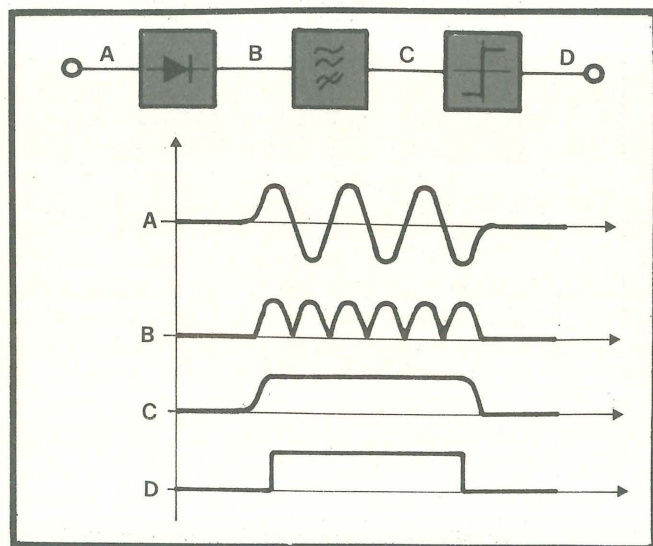



Figura 8 - Schema a blocchi di un demodulatore.

Figura 9 - Schema a blocchi del demodulatore a finestra.

opportune interfacce (vedi per esempio BIT n. 2) o collegandosi direttamente alle testine, è possibile arrivare, su registratori di buona qualità, a 9600 bit/sec. Generare segnali PAM (Pulse Amplitude Modulation) non presenta particolari difficoltà; la sezione trasmittente dell'interfaccia è semplicemente un oscillatore controllato dal bit SO (Serial Out) della porta di uscita. Si ha emissione di nota in corrispon-

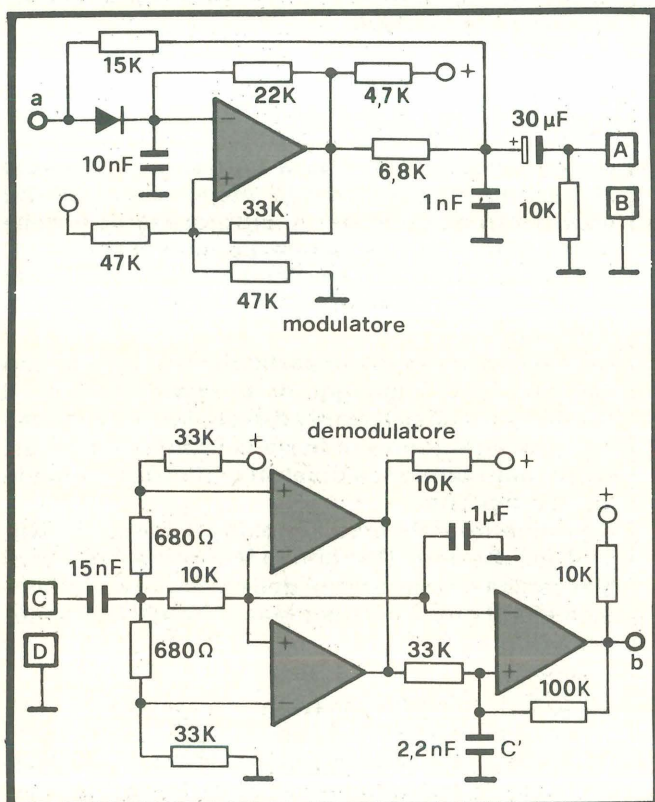



Figura 10 - Schema elettrico dell'interfaccia completa.

```
                              CASS_0                                   PAGE   1
LOC    OBJ CODE M STMT SOURCE STATEMENT                                 ASM 5.8

                    1  ;          PROGRAMMA PER LA GESTIONE DELLE
                    2  ;          CASSETTE MAGNETICHE
                    3  ;
                    4  ;
                    5  ;
                    6  ;
                    7  ;          DEFINIZIONE PARAMETRI
                    8  ;
                    9  ;
                   10  CONADD   EQU      0C00H    ;INDIRIZZO CONSOLLE
                   11  MAXBY    EQU      20H      ;MASSIMA LUNGHEZZA STRINGHE DI S
                   12  DELY     EQU      4AH      ;RITARDO TBIT/4
                   13  PRENUM   EQU      0F0H     ;DURATA PREAMBOLO
                   14  SI       EQU      6        ;BIT DI INGRESSO SERIALE
                   15  SO       EQU      7        ;BIT DI USCITA SERIALE
                   16  BITNUM   EQU      8        ;NUM. BIT PER CARATTERE DELL'UAR
                   17  ;
                   18  ;
                   19  ;
                   20  ;
                   21  ;
                   22  ;          AREE DI MEMORIA RAM
                   23  ;
                   24  ;
0BF4               25  RAMST    ORG      0BF4H
                   26  ;
0BF4               27  CKBUFF   DEFS     1        ;BUFFER CHECKSUM
0BF5               28  XX       DEFS     5        ;USATI DA PICOMON_0
0BFA               29  FROM     DEFS     2        ;IND. INIZ. PER SAVE
0BFC               30  TO       DEFS     2        ;IND. FINALE PER SAVE
0BFE               31  ABUF     DEFS     2        ;IND. AUTOSTART
                   32  ;
                   33  ;
                   34  ;
                   35  ;
0100               36  ROMST    ORG      100H     ;QUESTO MODULO SI ACCODA A PI
                   37  ;
                   38  ;
                   39  ;          GESTIONE DELLE
                   40  ;          CASSETTE MAGNETICHE
                   41  ;
                   42  ;          TRASFERISCE SU NASTRO IL BLOCCO DI MEMORIA
                   43  ;          DA (FROM) A (TO), CON AUTOSTART A (ABUF)
                   44  ;          CON PREAMBOLO
                   45  ;
0100   E5          46  SAVE     PUSH HL
0101   CD3501      47           CALL PREAMB       ;BLOCCO DI PREAMBOLO
0104   2AFC0B      48  SAVE1    LD HL,(TO)        ;INDIRIZZO FINALE => HL
0107   ED5BFA0B    49           LD DE,(FROM)      ;IND. INIZIALE => DE
010B   A7          50           AND A             ;CY = 0
010C   ED52        51           SBC HL,DE         ;CALCOLA LUNGHEZZA
010E   012000      52           LD BC, MAXBY
0111   ED42        53           SBC HL,BC         ;E' MAGGIORE DI MAXBY ?
0113   FA2501      54           JP M LAST         ;NO, E' L'ULTIMO BLOCCO
0116   EB          55           EX DE, HL         ;IND. INIZIALE => HL
0117   09          56           ADD HL,BC         ;IND. INIZ. BLOCCO SUCCESSIVO
0118   22FA0B      57           LD (FROM), HL     ;AGGIORNA FROM
011B   ED42        58           SBC HL,BC         ;IND. INIZ. CORRENTE BLOCCO => H

                              CASS_0                                   PAGE   2
LOC    OBJ CODE M STMT SOURCE STATEMENT                                 ASM 5.8

011D   112000      59           LD DE,MAXBY       ;LUNGHEZZA => DE
0120   CD4701      60           CALL SMSG         ;INVIA UN BLOCCO
0123   18DF        61           JR SAVE1          ;BLOCCHI SUCC. SENZA PREAMBOL
                   62  ;
                   63  ;
                   64  ;
                   65  ;          INVIA L'ULTIMO BLOCCO E L'AUTOSTART
                   66  ;
0125   09          67  LAST     ADD HL,BC         ;RICALCOLA LUNGHEZZA
0126   EB          68           EX DE, HL         ;LUNGH. => DE;IND. INIZ. => HL
0127   CD4701      69           CALL SMSG         ;INVIA UN BLOCCO
012A   110000      70           LD DE,0           ;LUNGH. = 0
012D   2AFE0B      71           LD HL, (ABUF)     ;IND. AUTOSTART => HL
0130   CD4701      72           CALL SMSG         ;INVIA BLOCCO AUTOSTART
0133   E1          73           POP HL
0134   C9          74           RET
                   75  ;
                   76  ;
                   77  ;
                   78  ;          INVIA UN PREAMBOLO
                   79  ;          0/1 ALTERNATI CON CADENZA TBIT/3
                   80  ;
0135   21000C      81  PREAMB   LD HL,CONADD
0138   06F0        82           LD B,PRENUM
013A   CBBE        83  PRELOP   RES SO,(HL)       ;INVIA "0"
013C   CD2102      84           CALL    RATED4
013F   CBFE        85           SET SO,(HL)       ;INVIA "1"
0141   CD1802      86           CALL RATEP2
0144   10F4        87           DJNZ PRELOP
0146   C9          88           RET
                   89  ;
                   90  ;
                   91  ;
                   92  ;          INVIA UN BLOCCO DI CARATTERI NEL FORMATO:
                   93  ;          1/0/LUNGH.L/H/IND.INIZ.L/H/DATI/CKS
                   94  ;
0147   EB          95  SMSG     EX HL,DE          ;IND. INIZ. => DE; LUNGH. => HL
0148   010600      96           LD BC,6
014B   09          97           ADD HL,BC         ;LUNGH. + 6 => HL
014C   3E00        98           LD A,0            ;AZZERA BUFFER CKS
014E   32F40B      99           LD (CKBUFF),A
0151   0E01       100           LD C,1            ;INVIA INTESTAZIONE
0153   CD0202     101           CALL SOUT
0156   0E00       102           LD C,0
0158   CD0202     103           CALL SOUT
015B   4D         104           LD C,L
015C   CD0202     105           CALL SOUT
015F   4C         106           LD C,H
0160   CD0202     107           CALL SOUT
0163   4B         108           LD C,E
0164   CD0202     109           CALL SOUT
0167   4A         110           LD C,D
0168   CD0...     111           CALL SOUT
016B   EB         112           EX DE,HL          ;LUNGH. => DE; IND. INIZ. => H
016C   7B         113  TXLOOP   LD A,E
016D   D606       114           SUB 6
...               115           OR D
```

Figura 11 - Listato del programma di gestione delle cassette magnetiche.

```
                               CASS_0                                   PAGE    3
LOC    OBJ CODE M STMT SOURCE STATEMENT                                 ASM 5.8

0172   4E          117          LD C,(HL)         ;INVIA BLOCCO DATI
0173   CD0202      118          CALL SOUT
0176   1B          119          DEC DE
0177   23          120          INC HL
0178   18F2        121          JR TXLOOP
017A   3AF40B      122  SCKS    LD A,(CKBUFF)     ;INVIA CKS
017D   4F          123          LD C,A
017E   CD0202      124          CALL SOUT
0181   C9          125          RET
                   126  ;
                   127  ;
                   128  ;
                   129  ;       USCITA SERIALE
                   130  ;       8 BIT + 2 DI STOP
                   131  ;       IL CARATTERE IN C ESCE SUL BIT SO
                   132  ;       DELLA PORTA CONADD
                   133  ;
0182   C5          134  SEROUT  PUSH BC
0183   E5          135          PUSH HL
0184   21000C      136          LD HL,CONADD
0187   060B        137          LD B,BITNUM+3
0189   CBBE        138          RES SO,(HL)       ;BIT DI START
018B   CB09        139  SLOOP   RRC C
018D   CD1B02      140          CALL RATE
0190   71          141          LD (HL),C         ;INVIA UN BIT
0191   CBF9        142          SET SO,C          ;PREPARA STOP BIT
0193   10F6        143          DJNZ SLOOP
0195   E1          144          POP HL
0196   C1          145          POP BC
0197   C9          146          RET
                   147  ;
                   148  ;
                   149  ;
                   150  ;       LETTURA DA CASSETTE DI UN BLOCCO
                   151  ;       NEL FORMATO DI SMSG
                   152  ;
0198   CD0902      153  LOAD    CALL SIN          ;CERCA INTESTAZIONE
019B   0D          154          DEC C
019C   20FA        155          JR NZ LOAD        ;PRIMO CAR. = 1
019E   CD0902      156          CALL SIN
01A1   79          157          LD A,C
01A2   A7          158          AND A
01A3   C23402      159          JP NZ ERR2        ;SECONDO CAR. = 0
01A6   3E01        160          LD A,1
01A8   32F40B      161          LD (CKBUFF),A     ;INIZ. BUFFER CHECKSUM
01AB   CD0902      162          CALL SIN
01AE   59          163          LD E,C
01AF   CD0902      164          CALL SIN
01B2   51          165          LD D,C            ;LUNGH. + 6 => DE
01B3   CD0902      166          CALL SIN
01B6   69          167          LD L,C
01B7   CD0902      168          CALL SIN
01BA   61          169          LD H,C            ;IND. INIZ. => HL
01BB   7B          170          LD A,E
01BC   D606        171          SUB 6
01BE   5F          172          LD E,A
01BF   B2          173          OR D              ;VERIFICA SE AUTOSTART
01C0   2001        174          JR NZ RXLOOP

                               CASS 0                                   PAGE    4
LOC    OBJ CODE M STMT SOURCE STATEMENT                                 ASM 5.8

01C2   E9          175          JP (HL)           ;E' AUTOSTART
01C3   CD0902      176  RXLOOP  CALL SIN          ;LEGGE BLOCCO DATI
01C6   71          177          LD (HL),C
01C7   23          178          INC HL
01C8   1B          179          DEC DE
01C9   7B          180          LD A,E
01CA   B2          181          OR D
01CB   20F6        182          JR NZ RXLOOP
01CD   CD0902      183          CALL SIN          ;LEGGE CHECKSUM
01D0   FE00        184          CP 0
01D2   C23802      185          JP NZ ERR3
01D5   18C1        186          JR LOAD
                   187  ;
                   188  ;
                   189  ;       INGRESSO SERIALE FORMATO ASINCRONO
                   190  ;       DATI DI 8 BIT + 1 STOP
                   191  ;       LEGGE IN C IL BIT SI DELLA PORTA CONADD
                   192  ;
01D7   E5          193  SERIN   PUSH HL
01D8   21000C      194          LD HL,CONADD
01DB   CB76        195  INTES   BIT SI,(HL)
01DD   20FC        196          JR NZ INTES       ;CERCA TRANSIZIONE 0 -> 1
01DF   CD1E02      197          CALL    RATED2
01E2   CB76        198          BIT SI,(HL)
01E4   20F5        199          JR NZ INTES       ;VERIFICA SE START BIT
01E6   0608        200          LD B,BITNUM
01E8   0E00        201          LD C,0
01EA   CD1B02      202  GETBIT  CALL RATE         ;LEGGE 8 BIT NEL REG. C
01ED   7E          203          LD A,(HL)
01EE   17          204          RLA
01EF   E680        205          AND 80H
01F1   81          206          ADD A,C
01F2   1F          207          RRA
01F3   4F          208          LD C,A
01F4   10F4        209          DJNZ GETBIT
01F6   CD1B02      210          CALL RATE
01F9   CB76        211          BIT SI,(HL)       ;VERIFICA BIT DI STOP
01FB   2833        212          JR Z ERR1
01FD   CB11        213          RL C
01FF   71          214          LD (HL),C         ;ECO SU VISUALIZZATORE
0200   E1          215          POP HL
0201   C9          216          RET
                   217  ;
                   218  ;
                   219  ;
                   220  ;       SCRIVE UN CARATTERE E AGGIORNA CHECKSUM
                   221  ;
0202   CD1002      222  SOUT    CALL CKS
0205   CD8201      223          CALL SEROUT
0208   C9          224          RET
                   225  ;
                   226  ;
                   227  ;
                   228  ;       LEGGE UN CARATTERE E AGGIORNA CHECKSUM
                   229  ;
0209   CDD701      230  SIN     CALL SERIN
020C   CD1002      231          CALL CKS
020F   C9          232          RET

                               CASS_0                                   PAGE    5
LOC    OBJ CODE M STMT SOURCE STATEMENT                                 ASM 5.8

                   233  ;
                   234  ;
                   235  ;
                   236  ;       AGGIORNA IL CHECKSUM IN CKBUFF
                   237  ;       USANDO IL CARATTERE IN C
                   238  ;
0210   3AF40B      239  CKS     LD A,(CKBUFF)
0213   A9          240          XOR C
0214   32F40B      241          LD (CKBUFF),A
0217   C9          242          RET
                   243  ;
                   244  ;
                   245  ;
                   246  ;       CADENZE PER UART E PREAMBOLO
                   247  ;
0218   CD1B02      248  RATEP2  CALL     RATE
021B   CD1E02      249  RATE    CALL     RATED2
021E   CD2102      250  RATED2  CALL     RATED4
0221   C5          251  RATED4  PUSH     BC
0222   064A        252          LD B,    DELY
0224   10FE        253  DELOP   DJNZ     DELOP
0226   C1          254          POP      BC
0227   C9          255          RET
                   256  ;
                   257  ;
                   258  ;       SEGNALAZIONE DI ERRORI
                   259  ;       NELLA LETTURA DA CASSETTE
                   260  ;       ERR N  FA COMPARIRE N SUL DISPLAY
                   261  ;
0228   21000C      262  ERR     LD HL,CONADD
022B   77          263          LD (HL),A
022C   C610        264          ADD A,10H
022E   18F8        265          JR ERR
                   266  ;
0230   3E01        267  ERR1    LD A,1            ;MANCA BIT DI STOP
0232   18F4        268          JR ERR
                   269  ;
0234   3E02        270  ERR2    LD A,2            ;ERRATA INTESTAZIONE
0236   18F0        271          JR ERR
                   272  ;
0238   3E03        273  ERR3    LD A,3            ;ERRORE DI CHECKSUM
023A   18EC        274          JR ERR
%
```

denza del livello 0; in questo modo le parti non registrate della cassetta sono interpretate come "linea a riposo" ed ignorate dall'UART *(v. Figura 4).*
Uno schema elettrico del modulatore è in *Figura 7.* L'oscillatore è un comparatore con isteresi reazionato RC, il valore di questi due componenti determina la frequenza della portante. Il diodo D blocca l'oscillatore forzando ad 1 l'ingresso del comparatore. Le resistenze R1 e 2 R1 formano un sommatore che elimina la componente continua in uscita nei periodi di riposo. Il filtro R' C' C" porta il segnale a valor medio nullo ed elimina le armoniche dell'onda quadra. Il valore di R' determina il livello del segnale in uscita.
Più complesso il problema del demodulatore, dal quale dipendono in massima parte le prestazioni del sistema. Occorre tener conto delle distorsioni introdotte dal canale (cioé dal registratore), quali rumore, limitazione della banda, variazioni di velocità e di livello.
Il demodulatore classico per segnali PAM si ottiene con un raddrizzatore a doppia semionda seguito da un filtro passa-basso ed un comparatore di soglia *(v. Figura 8).*
Il circuito qui proposto è una variante del raddrizzatore ad onda intera; non usa diodi e può funzionare anche con una singola alimentazione a 5 V. Lo schema a blocchi è in *Figura 9.* Il filtro passa-alto A elimina le fluttuazioni a bassa frequenza; il comparatore a finestra B rivela segnali di ampiezza superiore all'intervallo tra le soglie V_{R1} e V_{R2}. La sua uscita va all'allungatore di impulso D, che mantiene a livello alto l'uscita in presenza di impulsi ravvicinati provenienti dai comparatori. Uno schema elettrico completo è in *Figura 10.* Rispetto allo schema funzionale si ha un'inversione logica, perchè la presenza di nota corrisponde allo 0. Il comparatore a finestra è scisso in due circuiti, rispettivamente per la soglia superiore e per la soglia inferiore, le cui uscite (del tipo collettore aperto) sono direttamente collegate in OR cablato. Rallentando il tempo di salita con il condensatore C', si ottiene anche l'allungatore di impulso. ■

PICOCOMUNICATO

Le numerose richieste pervenute ci hanno portato a tempi più lunghi di quelli previsti inizialmente.
Se BIT da una parte intende favorire la diffusione della pratica dei sistemi a microprocessore, dall'altra non può farsi carico di tutto quello che un sistema di questo genere comporta: i tempi occorsi sono stati necessari per organizzare gli opportuni supporti tecnici e commerciali richiesti dalla diffusione del Picocomputer.
BIT continuerà a pubblicare articoli sul Picocomputer; comunque per ogni richiesta di informazioni o per chi è interessato all'acquisto del sistema occorre rivolgersi alla:
MESA 2 s.r.l. Via Canova 21 Milano - Tel. 02/3491040 - 316324

Caratteristiche del Picocomputer

Scheda di CPU

— *microprocessore:* Z80
— *frequenza di lavoro stabilizzata con quarzo:* 2,4576 MHz
— *EPROM 2716:* 2 Kbytes (di cui 250 bytes occupati da PICOMONITOR
— *RAM 2114:* 1 Kbytes
— *porta di Input parallela:* 1 da 8 bit
— *porta di Output parallela:* 1 da 8 bit
— *connettore:* a doppia fila di 37 contatti dorati passo 2,54
— *dimensioni scheda:* lunghezza 16 cm, larghezza 10 cm.
— *bus:* standard MUBUS
— *capacità di indirizzamento:* 64 Kbytes.
— *PICOMONITOR:* programma di monitor per la gestione della cassetta magnetica.

Tastiera

— *tasti:* 25 tasti professionali a scatto rapido
— *visualizzatore:* 8 display di grandi dimensioni - Altezza caratteri = 13
— *connettore:* cavo di tipo flat terminante con 2 connettori a 16 pin.

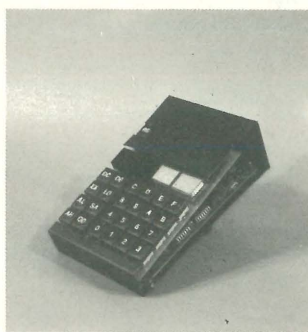
Cabinet

Supporto in alluminio per il contenimento della scheda di CPU e di tastiera - Sul cabinet è montato lo stabilizzatore a 5 volt con presa di alimentazione.

Alimentatore

Tipo a spina dotato di trasformatore, ponte e filtro.
Caratteristiche d'uscita: 10 V, 1A.

Prezzi del Picocomputer

Scheda di CPU	L. 120.000 (in Kit) L. 25.000 (stampato)
Tastiera	L. 50.000 (in Kit) L. 25.000 (stampato)
Cabinet	L. 10.000 (in Kit)
Alimentatore	L. 10.000
Interfaccia cassetta magnetica	L. 20.000 (in Kit)
Picocomputer con interfaccia cassetta magnetica assemblato e collaudato	L. 260.000 (senza alimentatore) L. 230.000 (senza alimentatore e senza interfaccia cassetta magnetica)

- I prezzi sono IVA esclusa
- Con la scheda e CPU è fornito il manuale di montaggio e/o d'uso
- Per ogni ulteriore informazione, rivolgersi alla ditta:
MESA 2 S.r.l. Via Canova, 21 Milano - Tel. 02/3491040 - 316324

Corso sul PASCAL
Struttura generale di un programma

di **F. Waldner** - *Università di Bari, Istituto di Fisica*

Parte III

In questo articolo cominceremo ad entrare nel vivo del linguaggio e parleremo della struttura generale di un programma nonché degli statements o istruzioni in PASCAL. Alla fine di questo articolo dovreste essere in grado di scrivere qualche programma in PASCAL, inizialmente senza l'uso di dati strutturati e senza imput/output. Sono, evidentemente, limitazioni che ridurranno i vostri primi programmi a poco più che esercizi accademici. Ciònonostante vi sarà utilissimo questo tipo di pratica: alla fine di questo articolo troverete alcuni suggerimenti per la stesura di alcuni programmi, con i quali potrete divertirvi un po'.

Procedure e funzioni

Questa è un po' una coda dell'articolo precedente: avevamo infatti parlato delle dichiarazioni e definizioni di labels, costanti, tipi e variabili. Ci mancavano le procedure e funzioni. In realtà non ne parleremo nemmeno ora, rimandando la cosa a quando parleremo esplicitamente di queste faccende. Per ora basterà osservare due cose: la prima è che le procedure (e funzioni) non sono altro che i sottoprogrammi in PASCAL (SUBROUTINES e FUNCTIONS nel FORTRAN); la seconda è che tutte costoro vanno definite e dichiarate nel programma che le usa: ciò è obbligatorio in PASCAL mentre non lo si fa mai (né lo si può fare) nel FORTRAN.
Una breve parentesi. Mi accorgo di parlare spesso del FORTRAN: non pensate che ciò succeda perché conosco solo quello (però un po' è vero). In realtà, il fatto è che il PASCAL è molto simile nella sua struttura all'ALGOL o al PL/1, e quindi chi programma in questi linguaggi dovrà sì cambiare la lingua, ma non la filosofia. Sono invece proprio i fortranisti quelli che dovranno fare gli sforzi maggiori per uscire dalla programmazione a spaghetti ed entrare nel mondo della programmazione strutturata.
Per altro mi accorgo, a questo punto, che ho messo in luce forse un po' troppo l'aspetto costrittivo in PASCAL, tralasciando quello vantaggioso. Cominciamo quindi a vedere quali sono i vantaggi di tutto il rigorismo che abbiamo visto fino ad ora.
A parte la programmazione più corretta, leggibile e disciplinata, c'è da notare un altro fatto: PASCAL è strutturato a blocchi, come ALGOL e PL/1, e non ha bisogno di cose come il COMMON del FORTRAN. Cos'è infatti il COMMON? È un'area di memoria *comune* (da cui il nome) fra i vari programmi, i quali sono liberi di accedervi, purché venga dichiarato che hanno bisogno di quell'area. PASCAL tira un frego su tutto ciò; non esistono più COMMON e similari, ed in compenso esiste qualche cosa di molto migliore: il programma è strutturato *a blocchi* o *ad albero* (block-structure, tree-structure), e per ciò stesso diventa un oggetto molto più razionale ed efficiente.
Vediamo dunque che cosa è una struttura ad albero: lo sa molto bene chi ha scritto in FORTRAN un programma per un mini da 16 bits, ed alla fine ha dovuto farlo lavorare in overlay, perché il programma era più lungo di 32K. In generale non è stata un'impresa facile. Bene: costui sarà felice di apprendere che nello scrivere il programma in PASCAL si scrive un programma *già pronto* per l'overlay.
Per tutti gli altri cominceremo dall'inizio. Supponete quindi di avere dei sotto-programmi variamente chiamati A1, A2,.., B1, B2,... etc., mentre il programma principale lo indichiamo con M (per *M*ain), e supponete che uno schema come in Figura 1 stia ad indicare la possibilità di chiamata delle varie procedure fra di loro.
Questa struttura, simile appunto ad un albero, sta ad indicare i collegamenti tra le varie procedure. Per esempio A2 è chiamata da M e può chiamare B1 e B2, però non può chiamare né A1 né B3: nella struttura sono infatti vietati i collegamenti orizzontali. Un altro modo di vedere la stessa struttura è "a blocchi" (v. Figura 2). M viene detto essere "al livello 0"; A1, A2, A3 "al livello 1"; B1, B2, B3 "al livello 2"; C1 "al livello 3". Ebbene: tutto ciò che è comunque definito in una procedura ad un livello, è considerato definito (e quindi accessibile) anche ai livelli superiori, purché le altre procedure siano legate alla prima o -se si vuole- facciano parte dello stesso blocco.
Se ciò suona un po' oscuro, la Figura 3 dovrebbe chiarirlo.
Esprimendo la cosa in un altro modo, osserveremo che B1 e B2 sono definiti quali procedure nel blocco A2 e per questo stesso fatto hanno accesso a tutto ciò che è comunque definito in A2, siano questi costanti, tipi, variabili strutturate e non.
Le dichiarazioni e definizioni hanno validità solo nel blocco in cui compaiono: questo viene anche chiamato lo *scopo* (scope) degli identificatori corrispondenti. Le dichiarazioni e definizioni che sono proprie

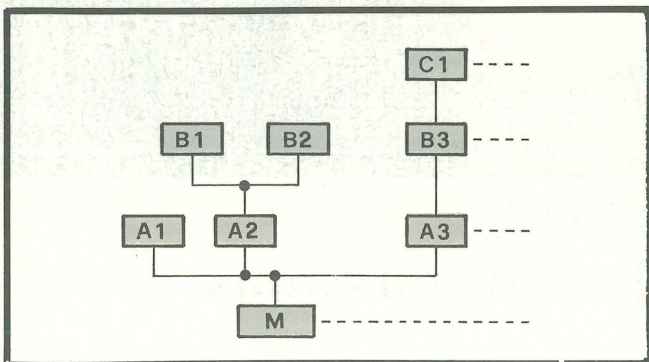



Figura 1 - Struttura ad albero di un programma. Le procedure a livello inferiore, possono chiamare quelle a livello superiore purché appartenenti al medesimo ramo.

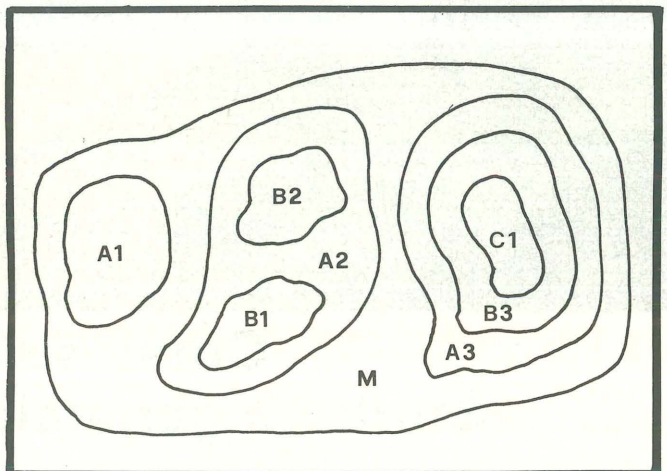



Figura 2 - Struttura a blocchi di un programma, in cui è evidenziato che quanto è definito a livello superiore è accessibile ai livelli inferiori, purché le procedure in questione facciano parte del medesimo blocco.

del blocco M appartengono a tutti i blocchi e quindi il loro scopo viene detto *globale* (global). Invece quelle che sono proprie per esempio del blocco A2 sono valide anche per B1 e B2, ma non per altri blocchi: il loro scopo è quindi detto *locale* (local).
Questo tipo di struttura permette di far funzionare un programma direttamente in overlay, come vi avevo detto. Nel nostro esempio, in memoria risiederà sempre il blocco M e -di volta in volta- A1 da solo, oppure A2 (con B1 e B2 che ne fanno parte a tutti gli effetti), oppure A3 (con B3 e C1).
Quindi, così come le procedure possono essere racchiuse una dentro l'altra, ciò vale anche per gli scopi degli identificatori. Questo "racchiudere" viene anche detto *nesting* con un termine figurato intraducibile, che originariamente voleva dire "lo star racchiuso come in un nido" (nest), e che poi è passato a significare "il rinchiudere una cosa dentro l'altra, come nelle scatole cinesi".

Gli statements di assegnazione

Cominciamo ora a vedere questi statements che sono simili a quelli di altri linguaggi, salvo alcuni dettagli, e sono fondamentali per il calcolo.
In generale la sintassi di uno statement di assegnazione è la seguente:

<statement di assegnazione>:: = <variabile> : =
<espressione>
(3.1)

che si legge "il valore della variabile è sostituito dal valore che risulta dal calcolo dell'espressione". Non confondete il simbolo di assegnazione (: =) con l'operatore relazionale (=). In PASCAL infatti scrivere A=B vuol dire scrivere una variabile booleana che è vera se A e B sono identici, altrimenti è falsa. Per il resto vedrete che le regole per scrivere le espressioni sono più o meno quelle a cui i fortranisti e algolisti sonoabituati: operatori, parentesi, variabili vi giocano più o meno gli stessi ruoli.
Vediamo in dettaglio come si scrive un'espressione. Ecco il BNF.

<espressione> :: = <espressione semplice> | (3.2)
<espressione semplice>
<operatore di relazione>
<espressione semplice>
<espressione semplice>:: = <termine>|
<espressione semplice>
<operatore di addizione>
<termine>
| <operatore di addizione>
<termine>

Tutto ciò che è definito nel blocco......	.....è anche definito nei blocchi.....
M	M, A1,A2,A3,B1,B2,B3,C1,
A1	A1
A2	A2, B1,B2
A3	A3, B3,C1
B1	B1
B2	B2
B3	B3, C1
C1	C1

Figura 3 -

<termine>:: = <fattore>|<termine>
<operatore di moltiplicazione>
<fattore>
<fattore>:: = <variabile>
|<costante non segnata>
| <designatore di funzione>
|(<espressione>)
| NOT <fattore>
<costante non segnata>:: = <numero non segnato>
| <stringa>
| <identificatore di costante>
<operatori di moltiplicazione> :: = *| /| DIV | MOD
| AND
<operatori di addizione> :: = + | - | OR
<operatori di relazione> :: = = | < | < = | <>
| > |> = |IN
<statements di assegnazione> :: = <variabile>: =
<espressione>
| <identificatore di funzione> : =
<espressione>

Detto così sembra molto complesso. Vedremo però che le cose non sono poi così tremende.
Esaminiamo le prime due della (3.2). Vediamo anzitutto che una espressione si può scrivere o come qualcosa detta "espressione semplice" (chiamiamola ES per il momento) o come *due* espressioni semplici, legate però tra loro da operatori di relazione (descritti nella VIII della (3.2)).
Per esempio, dette ES1 ed ES2 due espressioni semplici, un'espressione si può benissimo scrivere così:

ES1 > = ES2 oppure ES1 < > ES2 etc. (3.3)

Cos'è un'espressione semplice? Essa è definita in vari modi (vedete qui al lavoro il sistema di definizione top-down):
— o come un qualcosa che si chiama *"termine"*
— o come un termine preceduto da un operatore di addizione (definiti nella VII della (3.2))
— o come una serie di termini legati tra loro da operatori di addizione. Osservate come anche OR sia un operatore di addizione.

Supponiamo quindi che dei temini (non sappiamo ancora che cosa sono!) siano indicati con T1, T2, ...
Ecco varie possibilità di costruire espressioni semplici:

T1 (3.4)
T1 + T2
-T3
T4 OR T2
-T1 + T3 + T4 - T2

E continuiamo ad aprire scatole cinesi. Un *termine* è definito:
— o come un fattore
— o come una serie di fattori legati tra di loro da operazioni di moltiplicazione. E questi sono definiti dalla VI della (3.2). Vedete la presenza di AND e anche di DIV e MOD.

Quindi detti F1, F2,... dei fattori, ecco come si possono per esempio costruire dei termini:

F1 DIV F2 (3.5)
F3 MOD F1 DIV F4
F1 / F2* F3
F4 AND F3 AND F2

Le analogie con l'algebra (fattore = lettera o numero; termine = monomio; espressione semplice = polinomio) sono evidenti, anche se è bene ricordare che c'è un netto allargamento negli operatori di moltiplicazione e addizione.
E il gioco delle scatole cinesi è ormai finito. Un fattore può essere o una costante, o una variabile, o un'intera espressione, purché chiusa in parentesi, o una funzione, o un fattore preceduto da NOT.
Vediamo in pratica a che cosa siamo arrivati. Anzitutto abbiamo già stabilito una gerarchia delle operazioni: si calcolano *prima* i fattori, *poi* i temini, *poi* le espressioni semplici, *poi* le espressioni. Ciò vuol dire che nel caso di espressioni complesse si parte col calcolatore ciò che sta nella parentesi più interna (vi ricordo che un *fattore* può essere anche una intera espressione racchiusa in parentesi). A parte questo, tra tutti gli operatori viene data la priorità più alta al NOT logico. Solo dopo il programma esegue tutte le oerazioni *, /, DIV, MOD, AND. E per ultime vengono eseguite le +, —, OR.
Adesso rifacciamo, con gli esempi, il cammino inverso. Un fattore può quindi essere definito da:

— una variabile
    X, ABC, BEPPE, Z42, SKYLAB
— una costante *non* segnata
    3, 2.5, 3E-42, 'GIGI'
— un designatore di funzione
    SQRT (X), SIN (LATID/57.2958) (3.6)
— un fattore preceduto da NOT
    MARRIED NOT MARRIED (3.7)
    MALATO NON MALATO
— un'espressione in parentesi
    (SQRT(X) +3.47) (3.8)

Con i fattori possiamo costruire dei *termini*, legandoli con operatori moltiplicativi. Ecco alcuni termini:

```
4.25 * SQRT (X) / (3.27 + ABC)
(A +B) DIV (4 — C)                    (3.9)
STUDENT AND NOT SICK AND MARRIED
```

Ed infine le espressioni semplici, ottenute legando i termini con operatori di addizione (fra cui vi ricordo ancora l'esistenza di OR).
Al culmine della gerarchia, le espressioni, le quali possono essere di due tipi, come visto nella (3.3).

Ci sono al solito varie osservazioni particolari.
— Quando ci si trova di fronte a parecchi operatori moltiplicativi, questi vengono presi in considerazione seguendo l'ordine in cui appaiono.
Per esempio per scrivere

$$\frac{a}{bc} \qquad (3.10)$$

potete benissimo scrivere A/B/C. È però buona pratica non farlo, scrivendo invece:

A/ (B * C) (3.11)

Il risultato è naturalmente lo stesso, ma la comprensibilità è ben diversa.
— In generale abbondare di parentesi è una buona pratica che esalta la leggibilità. Un'espressione del tipo:

A * Z + C — Q * P >= D — 72 + D * MZ

(che dà come risultato un valore booleano) non è certo chiara come la:

(A * Z + C — Q *P) >= (D — 72 + D * MZ)

— In uno statement di assegnazione i tipi della variabile e della espressione *debbono essere uguali*. Questa è una restrizione ed una specie di passo indietro rispetto all'uso moderno. In realtà, mentre una volta questa legge veniva mantenuta soprattutto per avere compilatori semplici, ora è stata reintrodotta per avere una programmazione ordinata. C'è un'unica eccezione: se la variabile è reale l'espressione può essere sia reale che intera. In altri termini è lecito scrivere
<variabile reale>: = <espressione intera> (3.12)

e la conversione del risultato (intero) della espressione ad un numero reale avviene automaticamente. *Non* è però lecito il viceversa.
Viene quindi spontanea la domanda: se abbiamo una espressione reale (il cui risultato sia per esempio 4.3) come facciamo a convertire questo risultato in un intero? Ricordiamo subito che PASCAL non accetta espressioni come

N = 4.1 K = 4.7 (3.13)

(con N e R interi) che in FORTRAN assegnerebbero 4 (intero) sia ad N che a K. Vengono allora in aiuto delle "funzioni speciali" che connettono i tipi reali con i tipi interi.
Potremo scrivere per esempio:

```
N = ROUND (X)
K = TRUNC (Y)                    (3.14)
```

e notiamo che mentre la funzione TRUNC *tronca* la parte decimale di un numero reale, ROUND *arrotonda* il numero stesso.
Avremo così che:

```
ROUND (4.1)    dà 4          (3.15)
ROUND (4.8)    dà 5
TRUNC (4.1) }  danno 4
TRUNC (4.8) }
```

Oltre a queste due funzioni vi cito anche la funzione ODD (X) che, con X di tipo intero, ritorna un valore di tipo booleano. Più precisamente ritorna un valore *vero* se X è dispari, e *falso* se X è pari.
Siamo ora in grado di scrivere alcuni statements di assegnazione. Sceglierò alcune formule che tutti conoscono in modo che non ci siano problemi matematici a questo livello:

$S = \pi R^2$ (area del cerchio)
$V = 4/3\pi R^3$ (volume della sfera)
$D = \sqrt{b^2-4ac}$ (radice del discriminante di un'equazione di 2° grado)

divengono per esempio:

```
S : = PIGRECO * SQR (R)                    (3.16)
V : = (4/3) * PIGRECO * R * SQR (R)
D : = SQRT (B * B - 4 * A * C)
```

Con riferimento al volume della sfera, si può scrivere anche:

V := (4/3) * PIGRECO * EXP (3 * LN (R))

Vedete che l'assenza di un operatore di esponenziazione si fa sentire, mentre per il resto non c'è molto di diverso da FORTRAN o ALGOL.

Lo statement composto

Algolisti e similari non troveranno qui niente di nuovo, mentre per i fortranisti questa è una vera novità. Anzitutto vediamo il BNF:

```
<statement composto> ::= BEGIN                (3.17)
                         <statement>
                         {; <statement>} END
```

Perché è importante lo statement composto? Perché viene trattato come un unico statement, ma in realtà entro le parole BEGIN......END può trovare posto un intero programma, un'intera serie di statements. Questo porta immediatamente ad una drastica diminuizione dell'uso dei sottoprogrammi, ed è un passo concreto verso la programmazione strutturata.
Vediamo degli esempi. Sappiamo che nel caso in cui sia $b^2 - 4ac > 0$, le radici di una equazione di secondo grado sono:

$$x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \qquad (3.18)$$

Ebbene possiamo dare *due* radici con un *unico* statement (composto). Ed ecco come:

```
BEGIN
    D  := SQRT (B * B - 4 * A * C);          (3.19)
    X1 := (-B + D) / (2 * A);
    X2 := (-B -D) / (2 * A)
END
```

Così pure il seguente statement composto dà sia il volume che la superficie di una sfera di raggio R

```
BEGIN
    S  := 4* PIGRECO * SQR (R);              (3.20)
    V  := (4/3) * PIGRECO * EXP (3 *LN (R) )
END
```

Statements ripetitivi

Nel FORTRN sono di un solo tipo (la "DO"). In PASCAL ci sono invece ben tre tipi di ripetizioni possibili, che danno al programmatore una notevole flessibilità. Questi statements vengono chiamati: FOR, WHILE, REPEAT: li vedremo uno per uno qui di seguito.

Lo statement FOR

Spesso è noto a priori il numero di iterazioni che si devono eseguire: p. es. si devono moltiplicare due matrici fra di loro, oppure si deve controllare ed archiviare un certo numero di fatture, oppure si deve estrarre ed elaborare un certo numero di cartelle cliniche archiviate su nastro magnetico... In tutti questi casi conviene usare la FOR, che è quella che più si avvicina alla DO del FORTRAN. Nella FOR uno statement viene ripetuto più volte: una variabile di controllo parte da un valore iniziale e viene incrementata (o decrementata) di *uno* (e *solo* di uno, *non* di una quantità scelta dal programmatore) fino a raggiungere il valore finale. E se gli statements da eseguire sono più di uno? Niente paura: esiste lo statement composto che risolve tutto.
Eccovi il BNF:

```
<variabile di controllo> ::= <identificatore>
<valore iniziale>        ::= <espressione>
<valore finale>          ::= <espressione>      (3.21)
<statement FOR> ::= FOR
                    <variabile di controllo> :=
                    <valore iniziale>
                    TO <valore finale>
                    DO <statement>
oppure
<statement FOR>::= FOR
                    <variabile di controllo>:=
                    <valore iniziale>
                    DOWNTO <valore finale>
                    DO <statement>
```

Possiamo ora fare alcuni esempi. Supponiamo di indicare con "INPUT" un'operazione di input di un valore X. Vogliamo ottenere la media di 100 valori di X: ecco come possiamo procedere:

```
S:=0
FOR  N:=1  TO  100  DO
     BEGIN
       "INPUT";                              (3.22)
       S  := S + X
     END
S:=  S/100
```

Supponiamo ora di voler calcolare anche la deviazione standard della media. Bisognerà ricordare che, detta E(x) la media (*E*xpected value, cioé "valore atteso di x") ed $E(x^2)$ la media dei quadrati, si ha per la deviazione standard:

$$\sigma = \sqrt{E(x^2) - (E(x))^2} \qquad (3.23)$$

Il nostro programmino diviene quindi.

```
S1:=0;     S2:=  0;
FOR  N:=  1  TO  100  DO
     BEGIN
       "INPUT";                              (3.24)
       S1  := S1 + X;  S2:=  S2 + X * X;
     END
S1:=  S1 / 100;  S2:=  S2/100;
SIGMA  := SQRT  (S2 - S1 *S1)
```

Quindi nel nostro caso per la variabile di controllo è stato scelto il tipo intero. È necessario ciò? No, affatto: l'unica esclusione per la variabile di controllo è il tipo reale. Per esempio la variabile di controllo può essere del tipo CHAR: supponendo che i CHARACTER siano le sole lettere dell'alfabeto, ordinate in ordine alfabetico, e che siano definiti

```
ELLE  :='L';  ZETA  :='Z';                  (3.25)
```

potremmo avere statements del tipo

```
FOR  CH  :=  ZETA DOWNTO
             ELLE DO                        (3.26)
     BEGIN
     ............
     END
```

L'uso di questa possibilità nel maneggio di liste alfabetiche (cognomi di persone, etc.) è immediato. È ancora un po' presto però per vedere degli altri esempi concreti: occorrerà un po' più di materiale per poterlo fare.
Ultima particolarità; se avessimo uno statement del tipo:

```
FOR  K  := 4  TO  3  DO                     (3.27)
```

in cui (magari per errore) il limite superiore sia *più piccolo* del limite inferiore, il FOR *non* viene eseguito. Ciò contrasta col FORTRAN, in cui l'analoga DO

```
   DO  1  K = 4,3                           (3.28)
   ......................
 1 CONTINUE
```

viene eseguita una volta. Ciò avviene perché il FORTRAN effettua il controllo su K alla fine della DO. PASCAL effettua invece il controllo *prima* di cominciare ad eseguire la FOR. Il che -se vogliamo- è più corretto.

Gli statements WHILE e REPEAT

Così come il FOR è usato quando si conosce a priori il numero di iterazione da compiere, questi statements sono usati quando non è noto questo numero. È bene che li vediamo insieme, dal momento che -come vedremo- hanno funzioni analoghe.
Il BNF è semplicissimo:

```
<statement WHILE>::=WHILE
                    <espressione booleana>
                    DO<statement>
<statement REPEAT>::=REPEAT
                    <statement>
                    {;<statement>}
                    UNTIL
                    <espressione booleana>
                                            (3.29)
```

Penso che il BNF sia chiarissimo: WHILE dice in sostanza di eseguire uno statement (che può essere ovviamente del tipo composto) *fino a tanto che* (WHILE) è vera una certa espressione booleana scritta dal programmatore. REPEAT dice la stessa cosa. In che differiscono, allora? Differiscono nel fatto che mentre nel WHILE l'espressione booleana viene calcolata *prima* di eseguire gli statements, e quindi, se questa fosse falsa, gli statements non verrebbero eseguiti neanche una volta, ciò non succede nel caso di REPEAT, nel quale l'espressione booleana viene calcolata *dopo* l'esecuzione degli statements. Questi vengono quindi eseguiti almeno una volta.
È inutile -penso- farvi notare che tutti gli statements iterativi possono essere inclusi uno dentro l'altro (nesting). La programmazione con questo tipo di statements diviene molto più compatta e leggibile che non nel FORTRAN, in cui costruire una DO che venga eseguita fin tanto che è soddisfatta una condizione logica porta ad una codifica notevolmente involuta. Né aiuta molto l'uso (classico, al solito) di IF logiche e di GOTO: tutto ciò che si ottiene è -ancora una volta- la programmazione "a spaghetti".

Il GOTO

Questo è il residuo delle usatissime GO TO del FORTRAN e -come già vi ho detto- va usato il meno possibile. Al limite mai. Il BNF è semplicissimo:

```
<statement GOTO>::= GOTO <label>            (3.30)
<label>          ::={<digit>}^4_1
```

Vi ricordo che GOTO va scritta senza spazio in mezzo e che i labels vanno dichiarati.
Per esempio, se ad uno statement volessimo dare il label 13, dovremmo scrivere nella dichiarazione

```
LABEL  13                                   (3.31)
```

e lo statement in questione andrebbe scritto

```
13 : X:=   SQRT  (Y + Z);                   (3.32)
```

a questo punto una

```
GOTO  13;                                   (3.33)
```

farebbe saltare il programma allo statement 13 precedentemente definito.
Penso che non valga la pena di spendere più parole su questo argomento: può darsi che in qualche programma abbiate bisogno veramente di questo statement. Allora usatelo pure. Però meno lo userete e più sarete progrediti nella programmazione strutturata, nel metodo top-down e nel "pensare in PASCAL".

Gli statements condizionali

Questi sono un altro dei cardini di un linguaggio di programmazione. In PASCAL ce ne sono di due tipi: lo statement IF e lo statement CASE.
Vediamo anzitutto gli IF. Questi permettono di eseguire in sostanza uno statement oppure un altro se è verificata o meno una condizione logica. Ancora una volta vi ricordo che dire "uno statement" vuol dire anche -se necessario- "uno statement composto", cioè in pratica un vero e proprio sottoprogramma, e che il nesting è permesso anche nelle IF.
Ecco il BNF. Ci sono due tipi base di IF:

```
<statement IF (1)>::= IF                    (3.34)
                      <espressione booleana>
                      THEN < statement>
<statement IF (2)>::= IF
                      <espressione booleana>
                      THEN < statement>
                      ELSE < statement>
```

È evidente la funzione dei due tipi di IF: nel primo caso lo statement (anche composto!) viene eseguito solo se l'espressione booleana è vera, altrimenti viene ignorato. Nel secondo caso, oltre a questo succede che viene eseguito *un altro* statement nel caso che l'espressione booleana sia falsa. Attenzione: è proibito mettere un; prima di ELSE. Inoltre, forse sarà inutile per voi esperti, ma ve lo dico lo stesso: gli IF possono essere inclusi uno dentro l'altro col solito procedimento di nesting, ma state ben attenti a valutare quale IF conviene mettere più fuori nel nesting e quale più dentro: conviene mettere all'esterno le condizioni che sono verificate *più di rado*, in modo che il programma *quasi sempre salti tutto il nesting*, e mettere all'interno via via le condizioni più probabili.È sorprendente quanto tempo di calcolatore può venire risparmiato con un accurato esame di questo genere.
Finiamo questa parte con l'ultimo statement condizionale, lo statement CASE, definito col seguente BNF:

```
<statement CASE> ::= CASE                              (3.35)
                     <espressione non reale> OF
                     <elemento di lista CASE>
                     {;<elemento di lista CASE>}
                     END
<elemento di lista CASE>::=<lista dei labels CASE>;
                           <statement>|<nulla>
<lista dei labels CASE> ::= <label CASE>
                            { , <label CASE> }
<label CASE> ::= <costante>
```

Di che si tratta? Vediamo di spiegarci con degli esempi, e vi pregherei di controllare col BNF se gli statements riportati sono corretti.

```
CASE   K   OF                                  (3.36)
   1 : X:=3;
   2 : L:=L + 1;
   3 : Z =SQRT (Y);
   4 : BEGIN  K:= 3  K;  Y := 2  END
END
```

Uno statement di questo genere vuol dire che se K=1 viene eseguito X := 3, se K = 2 viene eseguito L := L + 1, e così via.
Ma ci possono essere delle situazioni più complicate. Per esempio:

```
CASE      K *K - 45  OF
3         : X := X + 2
29,32     : Z = Y + 3
-5,42, -3 : BEGIN H: = H + 1; Z = SQRT (X) END
END                                          (3.37)
```

In questo caso, se K * K-45 vale 29 o 32, allora viene eseguito lo statement Z := Y + 3, e così via.
Ma non è finita. Se CH è definita come carattere possiamo avere situazioni come

```
CASE  CH  OF                                   (3.38)
      'A' :           X:=-2
      'C', 'D', 'K' : BEGIN A:=3; B:=A+2 END
      'B', 'E'      :
END
```

In questo caso, a seconda di che cosa è CH (cioè a seconda di che lettera è), si hanno diverse scelte. Se CH vale A, viene eseguito X := -2; Se CH vale C o D o K, viene eseguito lo statement composto. Se vale B o E, non viene eseguito nulla: il programma procede semplicemente avanti.
È evidente che lo statement CASE è molto potente e versatile, e combinando tutto ciò per di più con il nesting, le possibilità divengono veramente incredibili.

Esempi di programmi

Passiamo ora a vedere alcuni esempi, che saranno necessariamente molto limitati in quanto non abbiamo ancora parlato di I/O, di dati strutturati, di tipi speciali. In sostanza ci dovremo ridurre a dei programmi di matematica semplice: sceglieremo allora la matematica che tutti conoscono.
Inizio con un programma per la soluzione di un'equazione di 2° grado. Vi ricordo che se $b^2 - 4ac < 0$ le soluzioni sono complesse coniugate, valendo la parte reale $- b/2a$ e la parte immaginaria

$$\frac{\sqrt{-(b^2 - 4ac)}}{2a}$$

Ecco qui allora un possibile programma:

```
      PROGRAM QUAD (INPUT, OUTPUT);
      {TO SOLVE QUADRATIC EQUATIONS}
      VAR FLAG: CHAR; A, B, C, X1, X2, D: REAL;
BEGIN D:=B * B - 4 * A * C; {DISCRIMINANT}
      IF D >= 0 THEN
        BEGIN    FLAG := 'R'; { REAL ROOTS}
          D := SQRT (D);
          X1 := (- B + D) / (2 * A);               (3.39)
          X2 := (- B - D) / (2 * A)
        END
      ELSE
        BEGIN    FLAG := 'I';

                                    {IMAGINARY
                                    ROOTS}
          X1:= -B/ (2 * A);
          X2:= SQRT (-D) / (2 * A)
        END                                        (3.39)
END
```

Quindi in X1 ed X2 troviamo le due radici, se FLAG = 'R'. Troveremo invece la parte reale e la parte immaginaria delle radici stesse se FLAG = 'I' e quindi le radici sono complesse. Prendiamo un altro esempio: nel calcolo combinatorio capita spesso di adoperare il cosiddetto "fattoriale", che è definito solo per interi positivi nel seguente modo:

$$n! = 1 \cdot 2 \cdot 3 \ldots (n - 1) \cdot n \qquad (3.40)$$

Inoltre si definisce 0! =1 con una definizione "ad hoc". Il nostro programma, quindi, dato un intero, dovrà ritornare un flag di errore se l'intero che gli abbiamo dato è negativo (in questo faremo in modo che il programma ci ritorni p. es.-1); dovrà ritornarci 1 se il numero dato è 0; dovrà eseguire il prodotto più sopra specificato se il numero dato è maggiore di 0.

Vediamo cosa si può fare: supponiamo che N sia il numero intero dato in qualche modo in ingresso.

Ecco come si può procedere:

```
IF N < 0 THEN  FACT := -1
ELSE                                          (3.41)
    IF N=0 THEN  FACT := 1
    ELSE {N > O  IN THIS CASE}
      BEGIN
      FACT := 1;
      FOR  K := 1 TO N DO
      FACT := FACT * K
      END
```

Vedete voi di completare le parti che mancano. Per la prossima volta potreste scrivere dei programmi che facciano la somma dei primi N numeri interi, dei loro reciproci (cioè $1 + 1/2 + 1/3 + 1/4 + ...$) e dei reciproci dei fattoriali $1 + 1/2! + 1/3! + ...$ Per quest'ultimo usate la formula $n! = 1 \cdot 2 \cdot 3........(n-1) \cdot n$ se $n \leqslant 30$ e invece la formula approssimata $n! =$

$\sqrt{2 \pi n}\, \frac{n^n}{e^n}$ se $30 < n \leqslant 50$.

Se n è più grande dì 50, allora trascurate pure il termine 1/n.
Questi programmi sono di puro esercizio: mi scuserete l'astrattezza dei temi, ma senza aver parlato ancora di dati strutturati e tipi speciali, non possiamo fare di più. ■

ERRATA CORRIGE

Nella Parte II (Bit n. 6) compaiono i seguenti errori:

Pag. 64. La metacostante CO è in realtà CONST.
Pag. 65. Nel CONTROLLO 3 il quesito I è in realtà: "Occorre dichiarare prima le costanti o le funzioni?".
Numerare con (2.4) e (2.5) i BNF di identificatore e blocco.
Pag. 66. Chiudere con apice il BNF di stringa.
Nel CONTROLLO 5 racchiudere tra apici i quesiti relativi a II, III, XVI.
Pag. 67. Nel CONTROLLO 7 il quesito III è: ..., KAPPA: CHAR; ...
Pag. 68. L'espressione (2.12) è $B \leqslant A$.
In Figura 2 INHIBIT è $X \cdot Y$; EXCLUSIVE OR è $\overline{X} \ominus Y$;
EQUALITY è $X \odot Y$.
Pag. 69. Correggere $1 < 1$ con $1 < 3$.
Nelle soluzioni (riquadro a pag. 69) correggere:
Nel CONTROLLO 2 X è giusto (invece che IV).
Nel CONTROLLO 6 III è errato perché un identificatore inizia con un digit.

Compucolor II. Grafici a 8 colori, prezzo in B/N.

Non a caso i professionisti si entusiasmeranno di fronte al Compucolor II.

È un sistema completamente integrato, basato sul microprocessore 8080A, con uno schermo grafico da 13 pollici a 8 colori programmabili, con minidisk da 51K per facciata e con l'interfaccia RS232C, il tutto già nella sua versione standard a un prezzo decisamente competitivo.

È programmabile in BASIC, ha 16384 punti indirizzabili sullo schermo e una presentazione di 32 linee per 64 caratteri di testo. La ROM da 16K contenente l'EXTENDED DISK BASIC consente un'accesso casuale ai FILES molto simile allo schema a memoria virtuale tipico dei grandi computers.

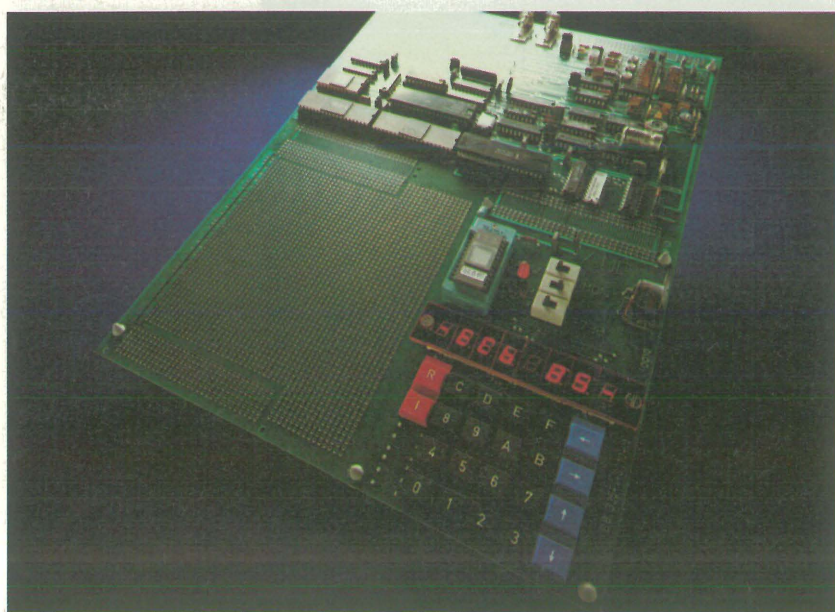
Le opzioni del Compucolor II sono costituite da ulteriori FLOPPY DISKS, dall'espansione da 16K a 32K della memoria RAM e da altri 2 tipi di tastiera.

DISTRIBUTORE PER L'ITALIA:
COMPITANT
VIA VITT. EMANUELE III, 9
91021 CAMPOBELLO DI MAZARA (TP)
TEL. (0924) 47153 - (0925) 72325

CONCESSIONARIO PER IL NORD-ITALIA:
SYMIC
MICROCOMPUTERS
& ELECTRONIC SYSTEMS S.R.L.
VIA PONTACCIO 12/a
20121 MILANO
TELEFONO 02/872414

CONCESSIONARIO PER
EMILIA E ROMAGNA, TOSCANA, MARCHE:
SORI S.N.C.
VIA BOLDRINI, 6
BOLOGNA
TELEFONO 051/558311

SIEMENS

ECB 85 per farvi dimenticare le limitazioni dei microcomputers didattici

Sotto la generica definizione di "microcomputers didattici" sono apparsi sul mercato in questi ultimi anni diversi sistemi a microprocessore.
Non sempre, tuttavia, le prestazioni fornite da tali sistemi rispondono in concreto alle aspettative degli utenti più qualificati.

Un microcomputer didattico sperimentale con caratteristiche avanzate
L'ECB 85 della Siemens possiede invece caratteristiche tali da permettere, mediante un apprendimento ordinato, un utilizzo autonomo, anche in campo progettuale, delle conoscenze acquisite.
Infatti su una sola scheda di 230x320 mm l'utilizzatore dispone di: un single board computer professionale con il microprocessore SAB 8085 • 44 linee di I/0 parallele • 2 contatori programmabili • 1+1/4 kbytes di memorie RAM e 2 zoccoli per memorie EPROM/ROM

• un interfaccia per registratore a cassetta • un programmatore EPROM • un'area libera per circuiti speciali • una tastiera a 22 tasti e display a 8 caratteri. Inoltre, l'ECB 85 lavora con una sola tensione di alimentazione (5 V) e un potente programma monitor implementa comandi di facile uso.

Come opzione, un system bus interface...
permette l'espansione dell'ECB 85 con moduli del sistema AMS 85 con la possibilità di realizzare configurazioni a multicomputer.

Una vasta area di applicazioni
L'ECB 85 serve come strumento didattico per tecnici e studenti, ma anche come sistema di sviluppo a basso costo, prototipo di laboratorio, microcomputer di controllo per macchine utensili ecc...

In italiano tutta la documentazione...
che non si limita a fornire le basi generali dei microprocessori, ma guida anche l'utente verso una serie di sperimentazioni sul sistema.

Per qualsiasi informazione, vi preghiamo di rivolgervi direttamente alla Siemens Elettra S.p.A., 20124 Milano, Via Fabio Filzi 25/A, Tel. (02) 6248
Divisione sistemi e componenti elettronici - Reparto A 201

dalla Siemens per il tecnico, lo studente, l'hobbista

Data Base Personale

Parte I **di F. Del Vecchio, M. Valsassina**

Introduzione

Il ritmo di vita aumenta, il dinamismo diventa frenetico, e la carta si accumula. Il medico si accorge di disperdere la propria professionalità compilando moduli, aggiornando schede e ricostruendo casi clinici. Il legale si arrovella nella giungla di atti, sentenze, leggi, decreti. Il manager necessita di informazioni sintetiche in tempo reale, senza dover personalmente interpretare numeri, dati, tabelle. L'impiegato si aliena nel lavoro impersonale legato ai carteggi. Diventa sempre più difficile gestire montagne di dati, labirinti di informazioni, e sempre maggiore risulta la necessità di delegare un lavoro così accurato, ma noiosamente ripetitivo, a chi sappia far presto e bene. Un valido aiuto oggi esiste: è il computer, una macchina logicamente tanto sofisticata ed in evoluzione da non poter essere ignorata.
Ogni applicazione di computers riguardante problematiche di gestione gravita intorno ad una organizzazione dati. Il DATA BASE, questo sconosciuto: possiamo immaginarlo come criterio di organizzazione delle informazioni e delle metodologie di utilizzo, così come in una oganizzazione di trasporti aerei esistono i mezzi e le rotte che ne permettono l'uso. Questo cardine della scienza informatica è stato ed è oggetto di impegnative ricerche, rese applicative nel campo E.D.P. dalla notevole richiesta di mercato. Già molto si è fatto, ed ogni elaboratore costruito per applicazioni gestionali è organizzato in modo da facilitare il compito del progettista di banche dati, sia dal punto di vista strutturale per l'accesso alle memorie di massa, sia per il software di base specializzato per queste applicazioni.
Ma *cosa si può fare* con un personal computer? Cosa permettono i naturali limiti di memoria e cosa si può trasferire dei concetti e soluzioni tecniche e adottati su computer di livello superiore? Questo è il nucleo dell'articolo, che si propone di chiarire, per quanto possibile, tali concetti con una proposta esempio di Data Base su personal. Riteniamo necessario rinfrescare prima alcuni principi teorici di organizzazione dati e considerare poi quali possono essere i conseguenti risvolti in ambiente personal computer.
Non desideriamo diffondere un articolo di opinione, consideriamo più importante l'aspetto istruttivo e costruttivo che una realizzazione pratica può offrire al lettore. Non sperate però di trovare qui la soluzione a tutti i vostri problemi, vi proporremo una struttura di gestione dati e il relativo programma BASIC completamente listato.
Per tale implementazione utilizzeremo l'Apple II, che con il DOS di buon livello di cui dispone, ci permetterà di sviluppare in modo interessante e chiaro i concetti base dell'applicazione. L'esempio non sarà limitato alla stampa di un listing, ma si procederà gradualmente dal discorso teorico alla stesura delle specifiche del prodotto desiderato, e successivamente all'analisi top-down e alla codifica; sempre permettendo al lettore di seguire l'intero evolversi del progetto, prima con ampie vedute dall'alto e poi sempre più particolareggiate, soprattutto sugli aspetti più interessanti.
Un impegno di questo tipo non ci ha però permesso di condensare tutto in un singolo articolo. Saranno necessarie più puntate, con sommo rammarico di chi vuole subito realizzare e concludere. In questo numero verrà presentato il discorso teorico e conoscitivo. Inoltre verranno definite le specifiche dell'applicazione da realizzare e, per la gioia di chi desidera qualche supporto pratico immediato, svilupperemo alcune routines che costituiscono strumenti essenziali per il management dei dati. Nel prossimo numero verrà completata la gestione del DATA BASE, con proposte e suggerimenti sulle possibilità di espansione e personalizzazione. Successivamente considereremo strumenti standard proposti e come poterne sfruttare le potenzialità.

Un po' di teoria

Il computer è composto da tre parti principali: uno strumento logico (CPU o unità centrale), la memoria con cui questo lavora e gli organi che colloquiano con il mondo esterno. La memoria direttamente accessibile dalla CPU è limitata per motivi di costo e dal tipo di unità centrale. Sono comunque disponibili memorie di massa di grossa capacità per sopperire a questa limitazione, come ad esempio dischi, nastri, ecc. Gli accessi in tal caso sono molto più lenti, perché l'elaborazione è possibile solo trasportando i dati da queste alla memoria interna. Da ciò nasce la problematica della gestione dei dati, cioé trovare il giusto compromesso tra costi, tempi e metodologia.
Iniziamo ora a vedere quali sono i principali problemi da affrontare nel progetto di una banca di dati, nei casi più generali; vedremo poi come applicare lo studio teorico al nostro caso, cioé all'applicazione su personal computer.
Il primo problema da porsi è come organizzare i dati, cioé che tipo di struttura logico-fisica si adatta meglio alle esigenze cui il nostro archivio cerca di sopperire. La *struttura fisica* caratterizza il modo in cui i dati sono fisicamente rappresentati nella memoria del calcolatore. La *struttura logica* invece, indipendentemente da quella fisica di memorizzazione, costituisce la rappresentazione dei dati come sono visti dall'utente del sistema. Le informazioni che memorizziamo in un archivio possono essere viste come gruppi di informazioni di base, tra i quali esistano delle relazioni logiche che li mettano in correlazione. La struttura logica di un archivio dice quali sono i dati elementari e i gruppi di essi, quali sono le relazioni tra

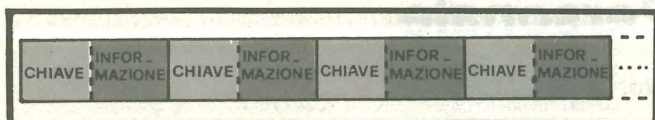



Figura 1 - Struttura sequenziale.

dati per formare i gruppi e quali sono le relazioni tra i gruppi che formano l'archivio.
Molto più complesso è il discorso per quello che riguarda la struttura fisica. La scelta di questa dipende prevalentemente dall'hardware a disposizione, dal tipo di elaborazione che si vuole sviluppare e dai tempi di accesso all'informazione necessari perché questa non perda di validità.
I parametri principali per la scelta di una struttura piuttosto che un'altra sono:

— il *coefficiente di densità*, dato dal rapporto tra lo spazio di memoria occupato dalle informazioni e lo spazio totale dell'archivio (comprensivo anche delle informazioni accessorie per il ritrovamento dei dati principali)
— la *lunghezza di ricerca*, cioé il numero di accessi necessari a reperire una informazione particolare
— la *facilità di aggiornamento* dell'archivio, cioé di cancellazione, inserimento o modifica di un dato.

I tipi più semplici di struttura fisica sono tre: *sequenziale, casuale* e a *liste*, e li vedremo ora per conoscerne pregi e difetti accenneremo nel seguito a strutturare di tipo più complesso.

strutture sequenziali

Questo è uno dei metodi più conosciuti ed usati per l'organizzazione dei dati: la caratteristica principale è che i dati componenti l'archivio sono memorizzati sequenzialmente uno dopo l'altro. Se pensiamo l'informazione suddivisa in due parti, attributo identificatore o chiave (codice associato ad ogni informazione che ne permette il reperimento), e informazione vera e propria, avremo un susseguirsi ininterrotto di coppie *(v. Figura 1).*

chiave informazione

Il problema della ricerca di una particolare informazione può essere risolto in vari modi:

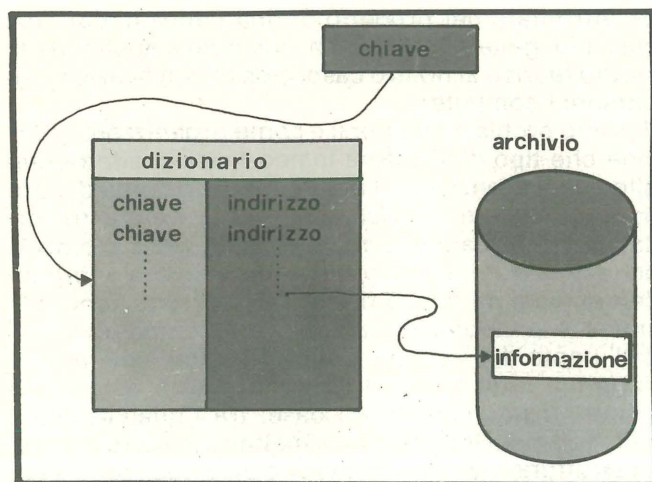



Figura 2 - Strutture casuali: metodo del dizionario.

Inizia da questo numero la pubblicazione di una serie di articoli sul DATA BASE, con particolare riferimento al personal computer. L'intento didattico nasce dalla necessità di rispondere in modo chiaro, sia a chi, con eccesso polemico, considera il personal computer solo come giocattolo (e si rifiuta di conoscerlo), sia a chi propone il confronto alla pari con i sistemi gestionali. Difficilmente è possibile configurare la soluzione tecnica appropriata in un mercato di prodotti dalla rapida evoluzione, ma il compito risulta impossibile quando l'informazione è drogata da promesse roboanti.
Per questi motivi è necessario arricchire il proprio livello culturale specifico; e la serie di articoli sul DATA BASE personale vuole dare la possibilità di conoscere quali sono i criteri base relativi al trattamento dell'informazione, e permettere di realizzare il significato pratico dei termini legati al dimensionamento del computer (Kbytes, CPU 8 bit, ecc.).
BIT propone una tematica educativa, che ci auguriamo venga recepita da tutti.
Non è un corso di DATA BASE, ma viene utilizzata enfaticamente tale denominazione per entrare nel mondo dell'elaborazione dati.

— *Ricerca completa:* si accede ad ogni registrazione, scandendole fino a trovare la chiave desiderata, o fino a scorrere tutto l'archivio. La lunghezza media di ricerca, se N è il numero di informazioni contenute nell'archivio, è (N+1)/2.
— *Ricerca seriale:* si richiede che l'archivio sia ordinato per valori crescenti della chiave; si accede sequenzialmente ai dati fino a trovare o la chiave cercata, oppure una chiave maggiore di essa. La lunghezza media di ricerca è N/2.
— *Ricerca binaria:* anche in questo caso si richiede che l'archivio sia ordinato. La ricerca avviene per successive bisezioni dell'archivio, per cui ad ogni passo si opera su un numero di dati pari alla metà del passo precedente. Si comincia a confrontare la chiave ricercata con quella che si trova a metà archivio: se questa è maggiore, si scarta la seconda metà dell'archivio, se è minore si scarta la prima metà. Al secondo passo si cerca la chiave che si trova a metà della nuova porzione di dati e si

Tabella I - Numero medio di accessi per trovare un dato con metodo HASH, in funzione del coefficiente di riempimento dell'archivio.

Coefficiente di riempimento archivio	Numero medio di accessi			
	metodo lineare	metodi casuale e quadratico	metodo del quoziente quadratico	metodo di concatenazione
0.1	1.06	1.06	1.05	1.05
0.5	1.50	1.44	1.39	1.25
0.75	2.50	1.99	1.85	1.38
0.9	5.50	2.80	2.56	1.45
0.95	10.50	3.45	3.15	1.48

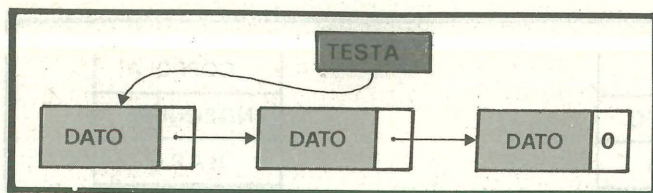



Figura 3 - Struttura a liste.

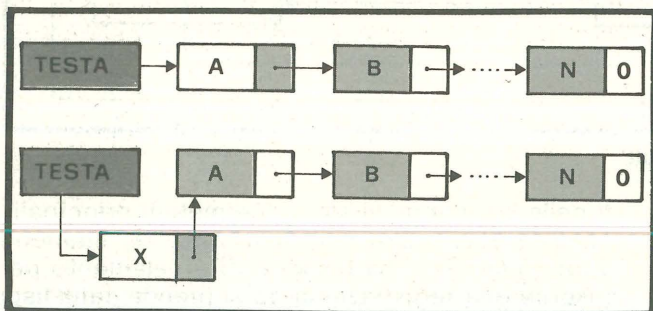



Figura 4 - La lista prima e dopo l'inserimento dell'elemento X.

procede come al passo precedente fino a trovare o no l'elemento. Questo metodo è molto veloce ed efficiente, perché il numero massimo di accessi per trovare una chiave o per stabilire che non esiste, è $\log_2(N)$: ad esempio per 1024 records il numero massimo è 10.

La struttura sequenziale presenta notevoli vantaggi di semplicità di gestione e di occupazione di memoria, non richiedendo informazioni supplementari per la ricerca del dato; con la ricerca binaria si ha anche un accesso molto veloce. Sono però un po' onerose le operazioni di aggiornamento, in quanto introdurre una nuova registrazione richiede un riordinamento di tutto l'archivio o un merge dei nuovi dati. Esistono comunque vari algoritmi molto efficienti di ordinamento (BUBBLESORT, HEAPSORT, QUICKSORT, ecc.).

strutture casuali

Le strutture casuali, cioé quelle nelle quali non c'é relazione tra gli indirizzi dei vari dati, ricavano direttamente dalle chiavi di accesso le informazioni sull'indirizzo dell'elemento:
esiste cioé una "funzione di accesso" che permette di ottenere l'indirizzo dalla chiave.
Esistono vari metodi di correlazione:

— *Metodo di dizionario:* permette l'accesso alla registrazione previa lettura dell'indirizzo in una tabella; cioé si crea una matrice di corrispondenza tra chiave ed indirizzo. Si cerca la chiave nella tabella e si reperisce anche l'indirizzo dell'informazione corrispondente *(v. Figura 2)*. È possibile anche la suddivisione dell'archivio in vari sottoarchivi, con la memorizzazione della chiave più alta di ogni sottoarchivio. La ricerca avviene prima nel dizionario, in cui si cerca il sottoarchivio di appartenenza del dato, e poi nel sottoarchivio con uno dei metodi visti per le strutture sequenziali.
— *Metodo di calcolo o indirizzamento Hash:* permette l'utilizzo di archivi di accesso diretto. L'associazione chiave-indirizzo viene realizzata mediante un algoritmo che estrae dalla chiave stessa l'indirizzo del dato relativo. Problema di questo metodo è di trovare un algoritmo che dia un risultato uniformemente distribuito, in modo da non avere sovrapposizione di record e di non avere grandi aree vuote.
Alcuni esempi di algoritmi HASH sono: l'indirizzo è dato da un sottoinsieme di bit della chiave; l'indirizzo è la somma, bit a bit, di diversi sottoinsiemi di bit della chiave; l'indirizzo è un sottoinsieme di bit del quadrato della chiave.
Scelto l'algoritmo di calcolo, si tratta di scegliere un algoritmo di scansione dell'area indirizzata, per gestire al meglio l'eventualità che due o più chiavi diano come risultato lo stesso indirizzo. I più diffusi sono:
— *La scansione lineare.* per cui in caso di collisione si scandisce l'archivio a partire dall'indirizzo trovato, sequenzialmente con un certo passo ≥ 1.
— *La scansione pseudo-casuale,* per cui gli incrementi di indirizzo rispetto a quello generato dall'algoritmo non sono costanti, ma sono prodotti da una routine che genera casualmente numeri interi.
— *La scansione quadratica,* metodo simile alla scansione lineare, solo che dopo ogni collisione per una data chiave, l'incremento è aumentato di uno (cioé alla prima collisione, l'incremento è 1, alla seconda è 2, ecc.). Variazione di questo metodo per aumentarne l'efficenza, è quello del quoziente quadratico, che consiste nel far variare l'incremento con la chiave invece di tenerlo costante.
— Il metodo di concatenamento per cui, trovato l'indirizzo con algoritmo, se esso è libero, si memorizza il dato, altrimenti si costruisce una catena (coda) di elementi relativi a quell'indirizzo. Vedremo in seguito cosa sono le catene.
Nella *tabella I* si vede l'efficienza dei vari metodi, in numero medio di accessi per trovare un dato in funzione del coefficiente di riempimento dell'archivio.

Figura 5 - La lista prima e dopo l'eliminazione dell'elemento X.

strutture a lista

Nella struttura a lista le registrazioni sono collegate tra di loro tramite puntatori, che in ogni elemento indicano l'indirizzo del successivo *(v. Figura 3)*.
La lista ha una testa che punta al primo elemento; ogni elemento punta al seguente, che può essere allocato ovunque nella memoria (di massa o interna del computer), non necessariamente nelle vicinanze;

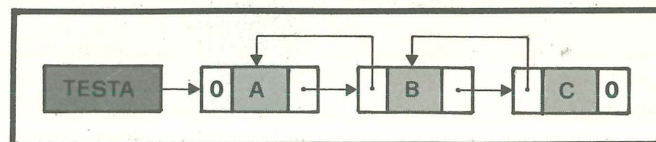



Figura 6 - Lista bidirezionale.

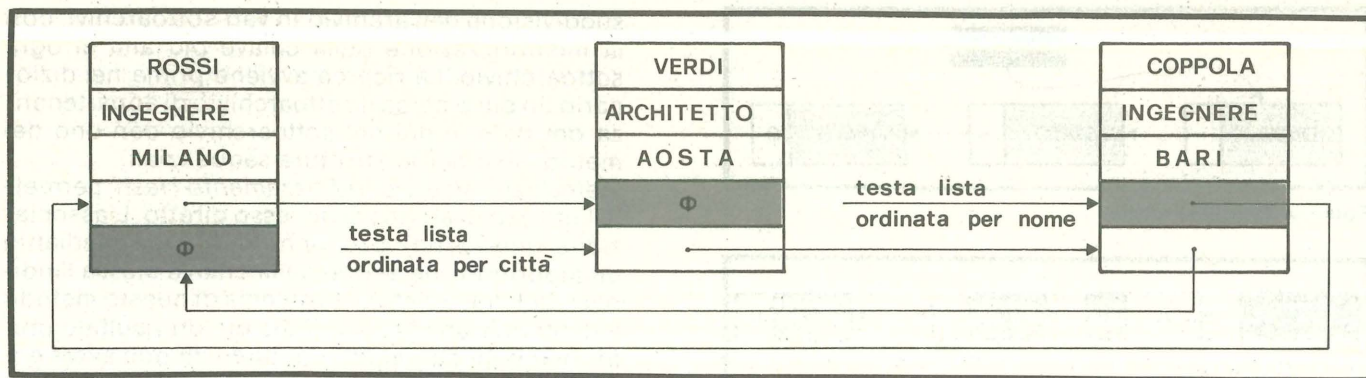



Figura 7 - Informazioni collegate in liste ordinate secondo alcuni attributi.

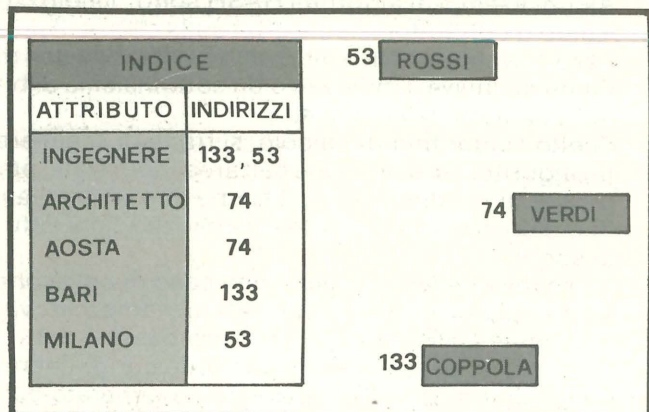



Figura 8 - Struttura a liste invertite.

l'ultimo elemento ha il puntatore uguale a 0.

Gli aggiornamenti di una lista sono abbastanza semplici:

— *Figura 4:* inserimento di un elemento X in una lista. Si è fatto l'esempio dell'inserimento in testa alla lista perché é il più veloce, ma anche un inserimento in un qualunque punto funziona nello stesso modo.

— *Figura 5:* cancellazione di un elemento X da una lista.

Un'ulteriore possibilità della struttura a lista è la lista bidirezionale, cioé in cui ogni elemento contiene 2 puntatori, uno all'elemento seguente e uno a quello precedente *(v. Figura 6).*

La ricerca di una particolare registrazione avviene scorrendo la lista della testa fino a trovare l'elemento desiderato. Per la gestione degli elementi cancellati e di quelli liberi per poter essere utilizzati nelle inserzioni, vi sono due metodi principali:

— *La lista libera:* è una lista formata da elementi vuoti; quando si ha bisogno di un elemento per inserire una registrazione, lo si prende dalla lista libera; ogni elemento cancellato viene inserito in essa, in modo da poter essere riutilizzato.

— *Garbage collection:* gli elmenti cancellati vengono trascurati; periodicamente un programma esamina tutta la memoria dell'archivio e riunisce questi elementi non utilizzati in una lista.

La struttura a lista ha il difetto che è necessaria della memoria supplementare per i puntatori, e che il tempo di ricerca è strettamente dipendente dalla lunghezza della lista.

strutture più complesse

Per problemi che non hanno una efficiente soluzione con le strutture ora viste, ne sono state studiate di più complesse.

— *Strutture a liste invertite:* se si volessero raccogliere gruppi di informazioni in liste ordinate secondo diversi parametri *(v. Figura 7),* si avrebbe uno spreco di memoria, in quanto sarebbe necessario un puntatore per ogni tipo di lista. Un metodo migliore è quello di definire per ogni elemento (nell'esempio per ogni persona) un insieme di attributi (titolo di studio e città); si crea poi un indice *(v. Figura 8)* contenente i valori degli attributi e i valori corrispondenti ad ogni elemento che ha quell'attributo, cosicché ogni elemento è ridotto ad una registrazione unica ad un certo indirizzo. A questo punto è possibile una ricerca delle regi-

Figura 9 - Strutture ad anello.

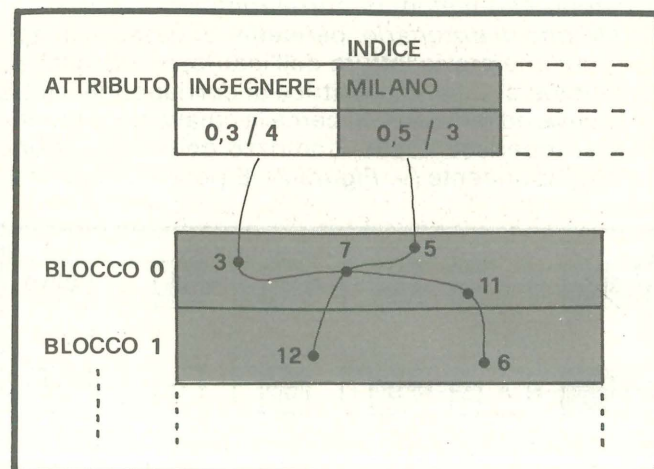



Figura 10 - Strutture a liste multiple.

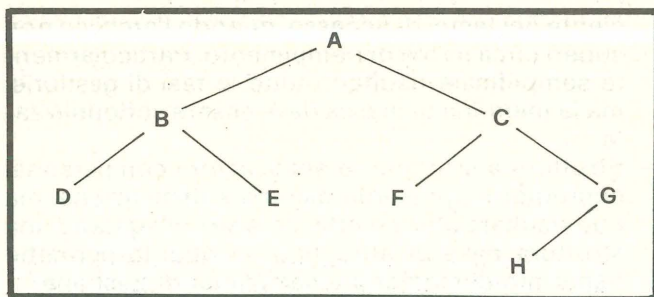



Figura 11 - Struttura ad albero.

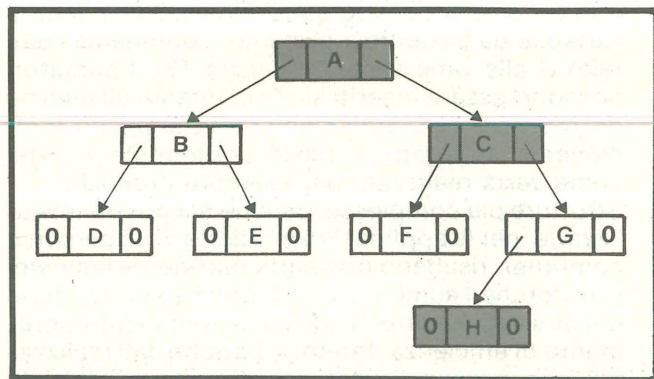



Figura 12 - Una possibile rappresentazione interna dell'albero di Figura 11.

strazioni secondo uno qualunque degli attributi definiti, utilizzando puntatori inseriti nella tabella indice. Il vantaggio delle liste invertite è che esse consentono di accedere a tutti i dati con la stessa facilità e con una maggiore velocità, in quanto si effettuano le ricerche solo sugli indici e poi si accede direttamente alla registrazione cercata.

— *Strutture ad anello:* sono un'estensione delle liste semplici, in cui l'ultimo elemento, invece di avere puntatore nullo, punta al primo elemento della lista. Permettono anche ramificazioni di una qualsiasi registrazione per individuare altri dati correlati ad essa *(v. Figura 9),* costruendo strutture gerarchiche. Queste strutture permettono di rappresentare relazioni complesse e la ricerca dei dati da un qualunque punto dell'archivio.

— *Strutture a liste multiple:* ricordano molto le uscite invertite, solo che in corrispondenza degli attributi sono memorizzati solo l'indirizzo di inizio della sottolista e la lunghezza di essa; inoltre la memoria in cui sono registrate le sottoliste è suddivisa in blocchi *(v. Figura 10).* Ogni registrazione è indirizzata dal numero del blocco e dall'indirizzo all'interno di esso. Quando si ricerca un elemento, si recupera con un solo accesso l'intero blocco, per cui si può avere risparmio di tempo, ed è quindi preferibile memorizzare i dati, per quanto possibile, in uno stesso blocco. Nella figura 10 si vede un esempio in cui le liste relative agli attributi INGEGNERE e MILANO sono lunghe rispettivamente 4 e 3, ed iniziano l'una nel blocco 0 con l'elemento 3, e l'altra, sempre nel blocco 0, ma con l'elemento 5. Inoltre si vede che possono esservi elementi come il 7, comuni a più liste. Le liste multiple, rispetto a quelle invertite, hanno il vantaggio di essere di più facile aggiornamento e programmazione, ma comportano tempi più lunghi di ricerca per lo scorrimento delle liste.

— *Strutture ad albero:* sono molto utili quando necessita che i dati siano organizzati in modo gerarchico. Un albero può essere definito come un grafo connesso privo di circuiti chiusi, composto da un certo numero di nodi (le informazioni) collegati da linee, con la proprietà che due nodi qualsiasi sono connessi da un unico cammino e che un albero con N nodi contiene N-1 linee. Il primo nodo è detto radice; un albero si dice binario se da ogni nodo escono al massimo due linee *(v. Figura 11).* Ogni nodo in un albero binario può essere rappresentato come l'insieme del dato più i puntatori dei due rami destro e sinistro *(v. Figura 12).* La scansione di un albero binario (visita di un albero) può avvenire in tre modi:

1) Ordine anticipato: esame della radice, esame del sottoalbero di sinistra, esame del sottoalbero di destra (nell'esempio l'ordine sarebbe ABDECFGH)
2) Ordine simmetrico: esame del sottoalbero di sinistra, esame della radice, esame del sottoalbero di destra (nell'esempio DBEAFCHG)
3) Ordine differito: esame del sottoalbero di sinistra, esame del sottoalbero di destra, esame della radice (nell'esempio DEBFCHGA)

Una struttura ad albero binario ha la proprietà che il sottoalbero sinistro di ogni nodo contiene solo dati la cui chiave è inferiore a quella del dato di quel nodo, mentre il sottoalbero destro contiene solo chiavi maggiori. La ricerca o la memorizzazione è abbastanza semplice, in quanto ad ogni nodo è immediato scegliere la via da seguire, fino a trovare il dato o un ramo libero in cui memorizzarlo.

E con un personal?

Il personal computer si compone di organi che hanno le stesse funzioni logiche di un computer tradizionale, cioé: unità centrale, I/O (Input/Output), memoria parte ad accesso diretto (RAM: Random Access Memory) e parte di massa. I costi contenuti e l'uso al quale il personal è diretto hanno orientato i costruttori nella scelta di CPU in genere ad 8 bit. Di conseguenza la memoria direttamente indirizzabile varia fra i 4 ed i 64 K bytes; considerando poi che una parte di questa è utilizzabile per funzioni interne (interprete, sistema operativo, ecc.), non rimane quindi memoria sufficiente per gestire il numero di informazioni necessarie per un Data Base in RAM, eccetto che per archivi minimi. Inevitabile la necessità di utilizzare supporti di memoria esterni, che quindi dovranno avere un accesso relativamente veloce, e soprattutto la logica di controllo dovrà permettere l'accesso diretto all'informazione elementare; altrimenti letture sequenziali penalizzerebbero drasticamente i tempi. Non consideriamo quindi l'ipotesi di utilizzare un supporto sequenziale come i nastri, ma soltanto i mini floppy disk, che riteniamo più validi per una applicazione informativa legata al Data Base. Nella maggior parte dei personal le unità a disco complessivamente collegabili possono ospitare fino a 500 K bytes circa in linea, cioè con tutte le informazioni direttamente accessibili, senza intervento manuale di cambio floppy.
Riprendiamo a questo punto i concetti teorici, per vedere quanto e con quali risultati siano applicabili ai personal.

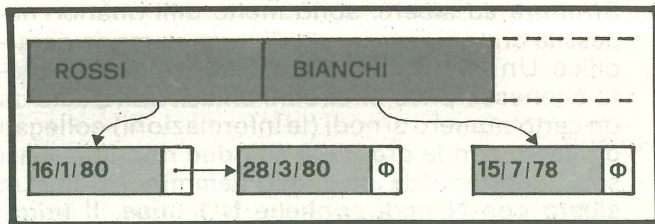



Figura 13 - Esempio di archivio di dati fissi, con collegate liste variabili di dati.

— Struttura sequenziale: essendo la più semplice, sembrerebbe la più adatta al nostro caso: in effetti, pur emettendo informazioni a lunghezza variabile, con conseguente impaccamento dei dati, i tempi di accesso appesantiscono l'elaborazione, e le scansioni prolungate sollecitano maggiormente l'unità di massa. Strutture di tale tipo sono preferibili per informazioni utilizzabili in blocchi, cioé una serie di dati che vengono letti, elaborati e scritti sempre tutti insieme.

— Struttura casuale: per i personal che permettono l'accesso diretto al record su disco, questa soluzione risulta ottimale, in quanto l'elaborazione necessaria per ottenere l'indirizzo dalla chiave non è assolutamente problematica, e l'accesso ad una singola informazione diventa quasi immediato. Il metodo del dizionario richiede maggiore spazio su disco per la memorizzazione delle coppie *chiave-indirizzo,* che possono essere tutte in memoria oppure su disco all'atto dell'elaborazione. Nel primo caso è necessario caricare tutto il dizionario in memoria all'inizio, ottimizzando così i tempi delle ricerche successive; nel secondo si può fare una ricerca binaria direttamente su disco, mantenendo il dizionario ordinato per *chiave.* Questa organizzazione, molto comoda nelle fasi di ricerca e modifica dell'informazione, è un po' più onerosa durante gli inserimenti e cancellazioni; perché necessita ogni volta la ristrutturazione del dizionario. Il metodo HASH, sebbene vi sia qualche problema per il trattamento del bit in BASIC, per il calcolo dell'indirizzo, è particolarmente efficiente nei tempi di accesso, quando l'archivio non superi circa il 75% di riempimento. Particolarmente semplificate risultano tutte le fasi di gestione, ma la memoria di massa deve essere sottoutilizzata.

— Struttura a liste: per le applicazioni con personal computer rappresenta già una sofisticazione, ma può risultare utile ed efficiente se collegata ad una struttura base di altro tipo, in quanto permette risparmio di memoria e semplicità di gestione. Ad esempio, nella costruzione di un archivio medico è possibile associare ad una serie di dati anagrafici fissi per ogni persona una lista a lunghezza variabile da paziente a paziente, contenente i dati relativi alle varie visite *(v. Figura 13).* I puntatori possono essere inseriti sia fisicamente all'interno del record, sia separatamente in file apposito, logicamente collegato a quello contenente le liste, come verrà realizzato nell'esempio proposto.

— Strutture più complesse: benché sia possibile realizzarle, per le applicazioni pratiche di un personal computer, risultano non particolarmente indicate. Ciò perché l'aumento di difficoltà nella gestione dei dati non è compensata da un sufficiente incremento di efficienza. Inoltre le banche dati realizzabili non possono avere una dimensione tale da giustificare una simile sofisticazione.

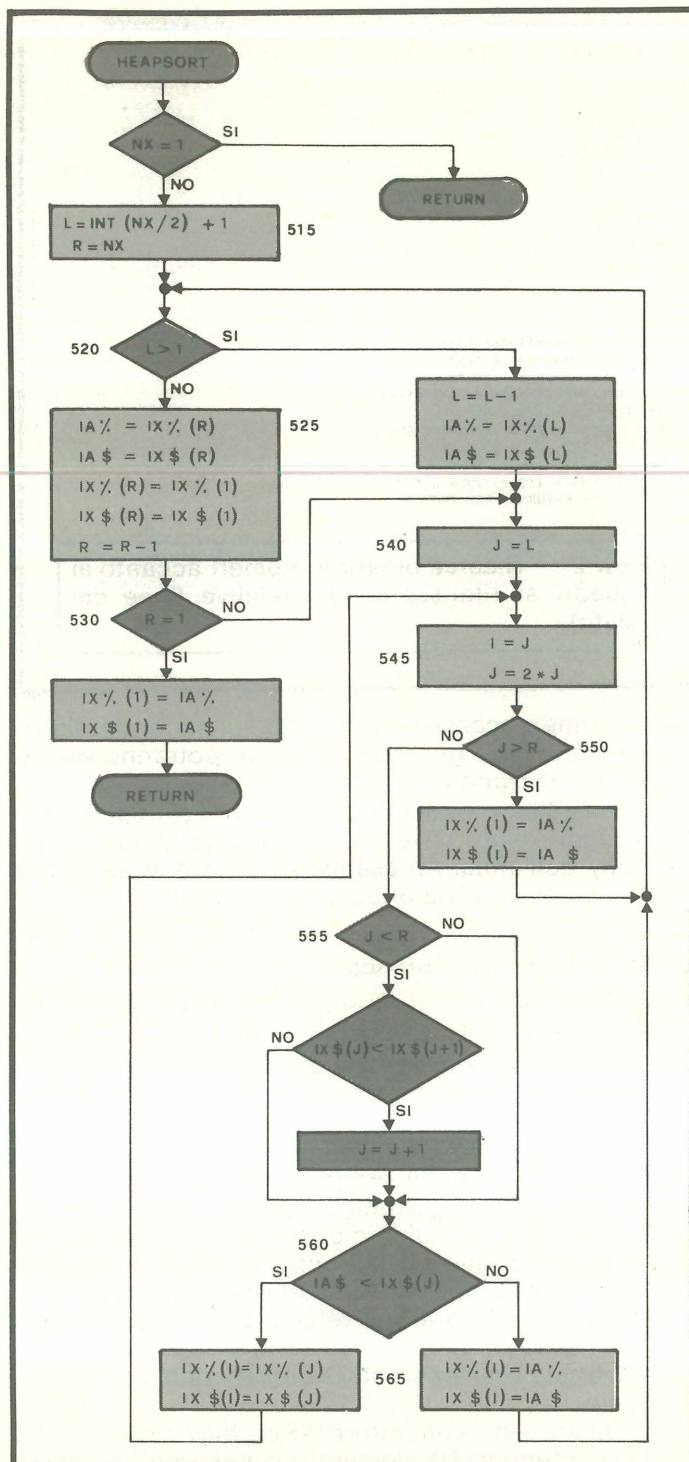
Figura 14 - Struttura generale del nostro archivio.

Definizione dell'applicazione

L'esempio applicativo vorrebbe essere di uso generalizzato, e soddisfare quindi più esigenze specifiche con un unico programma, studiato in modo da essere flessibile e adattabile.

Consideriamo alcune delle esigenze tipiche:

1) L'archiviazione delle cartelle cliniche consiste nella strutturazione di un archivio, costituito da gruppi di dati relativi ad ogni paziente di cui non si conosce a priori la lunghezza, in quanto la storia clinica può essere variabile. Il medico necessita di uno strumento che gli permetta di:
— inserire nuovi pazienti
— aggiornare la situazine
— conoscere tutti i dati patologici
— estrarre resoconti e situazioni sempre aggiornate.
2) Per la gestione vendite con agenti è necessario conoscere, oltre alle notizie generali sui volumi di vendita, anche informazioni più dettagliate relative ad ogni rappresentante. Inoltre interessante

La fotografia rappresenta una prova di riempimento di un archivio con tecnica HASH e scansione lineare con passo 1. La curva superiore rappresenta il numero medio di accessi al crescere del fattore di riempimento dell'archivio, e parte da 1 (un accesso solo ad archivio completamente vuoto) ed arriva a 7.8 ad archivio pieno. L'insieme di punti inferiore indica invece, per ogni record in ascissa, il numero di accessi che sono stati necessari per registrarlo.

```
500  REM *************************
501  REM * ROUTINE  ORDINAMENTO *
502  REM * HEAPSORT KNUTH VOL.3 *
503  REM *************************
510  IF NX = 1 THEN  RETURN
515 L =  INT (NX / 2) + 1:R = NX
520  IF L > 1 THEN L = L - 1:IA% = IX%(L):IA$ = IX$(L): GOTO 540
525 IA% = IX%(R):IA$ = IX$(R):IX%(R) = IX%(1):IX$(R) = IX$(1):R = R - 1
530  IF R = 1 THEN IX%(1) = IA%:IX$(1) = IA$: RETURN
540 J = L
545 I = J:J = 2 * J
550  IF J > R THEN IX%(I) = IA%:IX$(I) = IA$: GOTO 520
555  IF J < R THEN  IF IX$(J) < IX$(J + 1) THEN J = J + 1
560  IF IA$ < IX$(J) THEN IX%(I) = IX%(J):IX$(I) = IX$(J): GOTO 545
565 IX%(I) = IA%:IX$(I) = IA$: GOTO 520
```

Flow 1 — HEAPSORT (i numeri accanto ai riquadri si riferiscono alle relative linee del listato)

potrebbe essere conoscere statistiche di vendita per rappresentante e per articolo.

3) La gestione contatti con possibili clienti prevede la definizione delle caratteristiche anagrafiche e la possibiltà di seguire l'evolversi delle trattative e dell'acquisizione ordini.

Ed ancora analoghi discorsi per la gestione ordini, commesse, ecc. Tutte le applicazioni illustrate hanno in comune il fatto che le informazioni risultano composte da una parte fissa o di base alla quale sono agganciate informazioni accessorie. Nelle cartelle cliniche i dati base sono le informazioni anagrafiche, e quelle accessorie sono i risultati delle varie visite. Definiamo quindi di voler operare su di un archivio la cui struttura logica sia costituita da un insieme di gruppi di dati principali, ai quali possono essere associate una quantità variabile di altre informazioni.

Studio di fattibilità

Per poter fare uno studio di fattibilità del programma, è necessario conoscere il sistema su cui operiamo. L'Apple II EUROPLUS è dotato di un interprete basic floating point in ROM e di fino a 48 K bytes di memoria RAM, in cui risiede il programma utente e, se si utilizzano i mini floppy disk, il DOS, sistema operativo di gestione dischi, che occupa circa 10 K bytes. Sono collegabili fino a 12 unità floppy, ognuna delle quali può contenere 116 K bytes. Essendo il sistema operativo residente su ogni dischetto, sono disponibili per l'utilizzo 100 K bytes.

Questo ci impone una limitazione nel numero di records memorizzabili su ogni disco, e non volendo (benché possibile) gestire files multivolume, struttureremo l'archivio base su un unico floppy; il numero di records gestibili quindi sarà 100 K/lunghezza record. Avendo la possibilità di accedere in modo diretto ai records, per numero d'ordine, possiamo utilizzare per l'archivio base la struttura casuale col metodo del dizionario, costruendo un index che contiene tutte le chiavi di accesso e l'indirizzo relativo, dato come numero del record. Inoltre è possibile l'implementazione della struttura a liste; possiamo cioé costruire una o più liste facenti capo ad un record base. Il DOS 3.2 permette l'utilizzo di files senza doverne preventivamente definire le caratteristiche fisiche (come lunghezza record, spazio...), per cui è semplice usare files di descrizione della personalizzazione e di stato del sistema.

```
600  REM *************************
601  REM * ROUTINE RICERCA BIN. *
602  REM * SEARCH   KNUTH VOL.3 *
603  REM *************************
610 F = 0:L = 1:N = NX
620  IF N < L THEN  RETURN : REM NON TROVATO
630 P =  INT ((L + N) / 2)
640  IF KM$ < IX$(P) THEN N = P - 1: GOTO 620
650  IF KM$ > IX$(P) THEN L = P + 1: GOTO 620
660 F = 1: RETURN : REM TROVATO
```

Flow 2 — Ricerca binaria (i numeri accanto ai riquadri si riferiscono alle relative linee del listato).

Questa allocazione dinamica dello spazio ci permette una notevole flessibilità di configurazione, per cui possiamo scegliere tra diverse soluzioni.
La struttura globale *(v. Figura 14)* consterà di:
— Un archivio base, contenente dati fissi, non ordinati.
— Un index principale, contenente le chiavi e relativi indirizzi dei record base. Questo file dovrà essere mantenuto ordinato per chiavi crescenti.
— Eventuali index secondari per permettere l'accesso ai record base secondo altri campi.
— Uno o più archivi organizzati a liste legati logicamente ai record base.
A seconda delle particolari esigenze basterà personalizzare la configurazione, scegliendo il numero di drives da utilizzare. Cioé possiamo, se abbiamo pochi records base e non vogliamo index secondari e/o liste, allocare tutto l'archivio in un solo floppy. Per necessità maggiori si potrà riservare il primo disco all'archivio base e all'index principale, mentre le liste e gli index accessori andranno sul secondo disco. Oppure ancora gli index accessori potranno essere gestiti utilizzando un terzo drive. Avere le liste su un drive dedicato permette di avere più gruppi di liste per ogni record base, semplicemente cambiando il floppy dell'unità. Ad esempio è possibile associare allo stesso archivio di base liste mensili.

In riferimento al nostro articolo "CONFRONTO TRIDIMENSIONALE BASIC PASCAL" *apparso sul numero 6 di* BIT, *vorremmo dare una chiarificazione riguardante il listato* BASIC. *In esso appare il caricamento di due files binari,* "HI—RES CHARACTER GENERATOR" *e* "CHARACTER TABLE". *Sono due utility fornite dalla Apple Computer, e disponibili presso tutti i rivenditori assieme al manuale d'uso, che permettono di scrivere molto facilmente linee di testo sulla pagina di grafica ad alta risoluzione. Ci scusiamo con coloro che avessero avuto difficoltà nel comprendere il programma a causa di questa dimenticanza. In ogni caso il programma funziona anche senza l'uso di questi files: è sufficiente cancellare le linee da 2010 a 2060 e da 5200 a 5380, e sostituire la 5050 con la seguente linea:*
5050 HGR: HOME

Routines di utilità pratica

Nel corso dello studio teorico e di fattibilità, si è fatto cenno a necessità di ordinamento e di ricerca binaria. Anticipiamo in questo numero le routines che ordinano un vettore di N elementi e che realizzano in tale vettore la ricerca binaria. Abbiamo scelto due routines i cui algoritmi fondamentali sono dovuti al Knuth (The art of computer programming).
L'Heapsort è un algoritmo fra i più veloci, che presenta la caratteristica di impiegare un tempo quasi costante qualunque sia la configurazione iniziale del vettore. L'elaborazione si suddivide in due parti:
— una fase preparatoria veloce, in cui si costruisce un "heap"
— una successiva passata di ordinamento vero e proprio.
La routine ordina un vettore IX$ di chiavi (alfanumeriche) contenente NX elementi, mantenendo l'associazione con il corrispondente vettore IX% di indirizzi (interi). Il Flow non risulta strutturato, a vantaggio della velocità di esecuzione e dell'occupazione di memoria.
L'algoritmo di ricerca binaria, precedentemente descritto' opera sul vettore IX$, cercando in esso la chiave fornita in KM$ e lasciando in F il risultato della ricerca e in P l'indice dell'elemento, se esiste. ■

Bibliografia

J. K. Lion: *I data base. Organizzazione logica e realizzazione pratica.* Franco Angeli editore.
E. S. Page - L. B. Wilson: *Rappresentazione delle strutture informative e algoritmi di base.* ISEDI editrice.
C. Martella: *Impianti per l'elaborazione dell'informazione.* CLUP (cooperativa libraria università Politecnico di Milano)
D. O. S. VERSION 3. 2. (Apple Computer inc.)

Quel certo problema di informatica distribuita... prima o poi saremo noi a risolvertelo

Non siamo noi ad aver inventato l'informatica distribuita.
Ma siamo noi, oggi in Italia, quelli in grado di saperne tutto.
Tanto da potervi fornire qualsiasi **periferica, qualsiasi servizio inerente: dalla consulenza alla gestione, alla manutenzione,** all'assistenza.
Non ci conosci? Devi.
Informatica distribuita: prima o poi ci dovrai chiamare.
Perchè oggi, l'unica soluzione al tuo problema siamo noi.

segi. non ci conosci? devi.

20124 MILANO - Via Timavo, 12
Tel. (02) 6073184-6073255-6073088-692882
00199 ROMA - Via Asmara, 58
Tel. (06) 8395766

segi SERVIZI GENERALI PER L'INFORMATICA

informatica distribuita, tutto.

I "TANTI" PRODOTTI OK PER IL WIRE WRAPPING

wire wrapping center

Avvolge, Spella e Svolge filo di AWG 30 (0,25 mm) su terminali quadrati da 0,63 mm di lato

ATTREZZO "HOBBY WRAP"

Wrappatura regolare	WSU-30
Wrappatura modificata	WSU-30 M

ATTREZZO A BATTERIA modello BW-630

completo di Punta e Copri Punta PER TERMINALI di 0,63 mm di lato

PER AWG 30	BW-630
PER AWG 26/28	BW-2628

PUNTE DI RICAMBIO	BT-30
PUNTE DI RICAMBIO	BT-2628

INSERITORE DI DIP/IC

Raddrizza Pins — Rilascia — Ritiene — Inserisce

■ Modello INS-1416

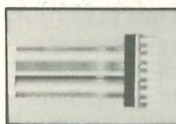
BOBINE DI FILO

Filo per Wire-Wrapping AWG 30 KYNAR
15 mt di conduttore solido ricoperto d'argento, facile da spellare.
Disponibile nei colori: Blue, Bianco, Giallo e Rosso.

DISPENSATORE DI FILO

- Spella automaticamente il filo per 25,4 mm
- Taglia Filo Incorporato
- Ricaricabile

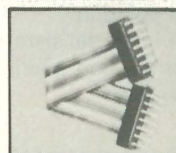
ESTRATTORE modello EX-1

Ideale per hobbisti e tecnici di laboratorio.
Può estrarre tutti i LSI-MSI-SSI da 8 a 24 Pin.

ZOCCOLI ASSEMBLATI da un lato solo con RIBBON CABLE

■ Cavo piatto da 26 AWG (0,40 mm) lungo 609 mm

Modello SE 14-24	con 14 Pin Dip Plug
Modello SE 16-24	con 16 Pin Dip Plug

KIT per Wire Wrapping modello WK-4B

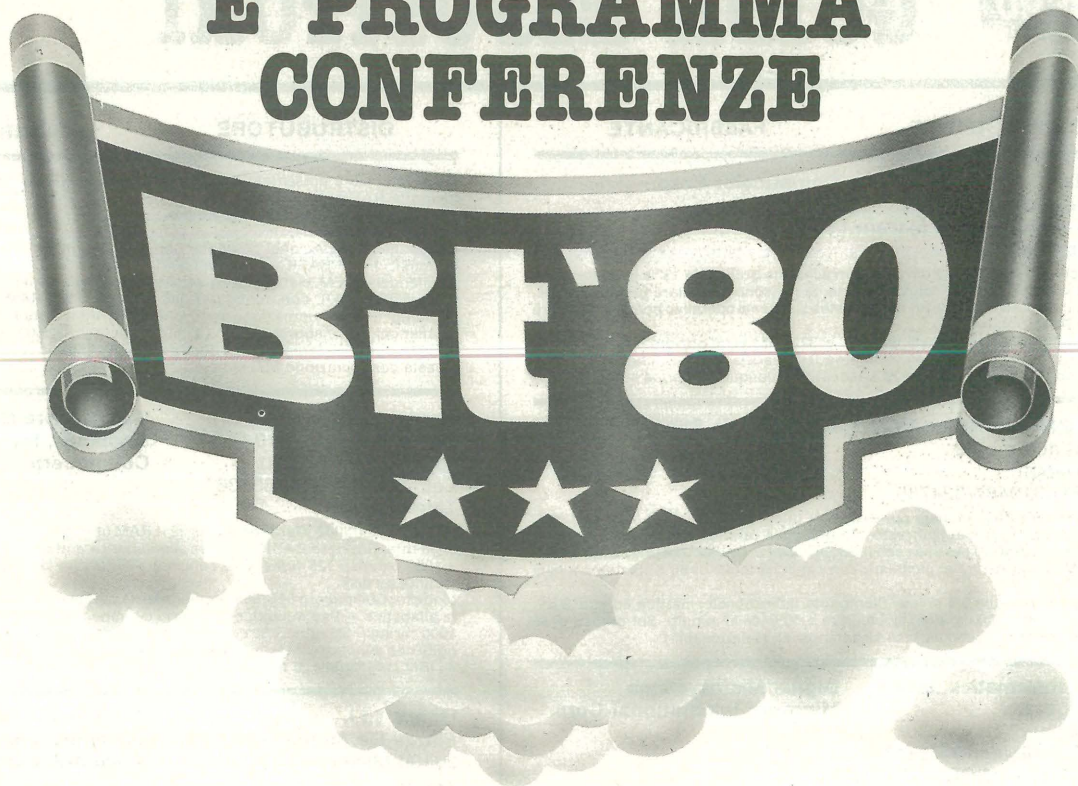
Contiene:
Attrezzo WSU-30 M
Dispensatore di filo WD-30-B
Due DIP da 14 Pin e due da 16 Pin
Una scheda hobby H-PCB-1
Un inseritore INS-1416
Un estrattore EX-1
Un connettore PCB modello CON-1

ZOCCOLI ASSEMBLATI con RIBBON CABLE da entrambe i lati

Con 14 Pin Dip Plug lungo 50,8 mm	DE 14-2
Con 14 Pin Dip Plug lungo 203,2 mm	DE 14-8
Con 16 Pin Dip Plug lungo 101,6 mm	DE 16-4

* Numerosi altri modelli a Stock

OK MACHINE & TOOL ITALIA S.R.L.

VIA CARLO RAVIZZA 34/1 - 20149 MILANO - TEL. 02/4983573 - TELEX 320419 BRUS I

GUIDA ESPOSITORI E PROGRAMMA CONFERENZE

Bit'80 ★★★

2ª Rassegna del personal & home computer e microprocessore 8/11 MAGGIO 1980

U.S. International Marketing Center
(Centro Commerciale Americano)
Via Gattamelata, 5/Milano-Fiera
Orario: 9,30/18
INGRESSO LIBERO

BIT '80 è organizzata dall'U.S. International Marketing Center e dal Gruppo Editoriale Jackson

GUIDA ESPOSITORI

DISTRUBUTORE	FABBRICANTE

Compitant — **Compucolor Corp.**
Via Vittorio Emanuele, 9
91021 Campobello di Mazara (TP)
Tel. (0924) 47153

Il Compucolor II è un microcomputer con schermo grafico da 13" a 64 colori con i punti 128 x 128 indirizzabili singolarmente. Si possono mescolare grafici e scritte. L'unità è completa di linguaggio BASIC esteso, sistema operativo per microfloppy e grafici su ROM (totale 24 K).
Viene distribuito in due modelli da 16 K e 32 K. Ha varie opzioni, come il floppy aggiuntivo e le tastiere speciali. Contiene l'interfaccia RS232C, che consente di trasformarlo in terminale oppure di attaccare qualunque periferica seriale.

Computer Company s.a.s. — **Tandy Radio Shack Corp.**
Via S. Giacomo, 32
80133 Napoli
Tel. (081) 310487/324786

Microprocessore Z80. Tastiera professionale integrata 53 tasti più tastiera numerica. Display: memory mapped; testo 16 righe di 64 caratteri con il BASIC level 1 e l'opzione di 32 caratteri per linea con il BASIC level 2. Grafici: 128 orizzontali per 48 verticali. Memoria centrale: modulare con capacità totale di 60 K. Sistema a mini disk:
interfaccia di controllo a 4 drives. Stampante bidirezionale: matrice ad aghi 9 x 7; velocità di stampa 120 caratteri al secondo e 50 linee al minuto. Set grafico di 96 oppure 160 caratteri, sia maiuscoli che minuscoli (selezionabili).

Condor Informatics Italia — **Digital Microsystems**, **Micropro International Corp.**
Via Podgora, 7
20122 Milano
Tel. (02) 5455809/5455813
Telex 334229 DECS I

La Micropro International Corp. è stata fondata circa due anni fa per sviluppare prodotti a microprocessore di quaità professionale sia per l'aspetto hardware che per quello software. Attualmente la Micropro presenta il WORDSTAR, un sistema di word processing di notevole fattura, adattabile a qualsiasi lingua europea e compatibile con sistemi basati su 8080, 8085, Z80 e dotati di CP/M.
Altri prodotti presentati sono: terminali video, stampanti ad aghi e a margherita, sistemi a minifloppy, a floppy e a disco con linguaggi di programmazione quali BASIC e Cobol ANSI 74.

Contradata Milano s.r.l. — **Data Specialties, Inc.**, **Decitek**, **MFE Corporation**
Via Valtortà, 11
20127 Milano
Tel. (02) 2892973/2828882

Floppy disk drivers doppia faccia, singola o doppia densità, a trazione diretta con motore a corrente continua, testine di tipo flottante che eliminano i problemi di usura del dischetto.
Tape cassette drivers, tape cassette terminals, paper tape reader/punchers.

De Mico Giuseppe s.p.a. — **Rockwell International, Inc**
Via Manzoni, 31
20121 Milano
Tel. (02) 653131
Telex 312035

L'AIM 65 è una pistra a microprocessore completa di: stampante termica ASCII 20 colonne, display 20 caratteri ASCII, interfaccia per due audiocassette e TTY, tastiera alfanumerica (54 tasti), 1 K o 4 Kbytes RAM, bus espandibile esternamente. Monitor interattivo da 8 K in ROM.
Altri prodotti presentati sono sistemi a microprocessore per office automatic ed interfacce per l'industria elettronica.

Edelektron s.r.l.
Corso Sempione, 39
20145 Milano
Tel. (02) 3493603/3185678/3185571

Documentazione tecnica della Intel, calcolatori H.P. e T.I., Cosmac Evaluation Kit, microcomputer su scheda TM 990/189.

Ediconsult s.r.l. — **Altos Computer System**, **Intertec Data Systems**, **North Star Computers, Inc.**, **Ohio Scientific, Inc.**
Via Rosmini, 3
20052 Monza (MI)
Tel. (02) 383950/360727

I prodotti Atlos comprendono la serie ACS 8000-X, costituita da calcolatori utillizzabili per applicazioni gestionali/scientifiche. Caratteristica fondamentale è la modularità, che permette di configurare il sistema secondo varie esigenze. L'hardware della serie ACS 8000-X comprende: microprocessore Z80, memoria RAM (da 32 K fino a 208 K), floppy disk (da 14,5 Mb o da 29 Mb). Terminali video: è possibile collegarne fino a 4 operanti indipendentemente. I linguaggi di programmazione disponibili sono: Assembler, BASIC, Fortran, Pascal, Cobol.

Dalla Internec è presentato il microcomputer Superbrain (microprocessore Z80, memoria RAM espandibile fino a 64 K) fornito di video (1920 caratteri) ed unità minifloppy con capacità di 360 Kbytes.
I prodotti Ohio sono:
Challenger 1-P, personal computer con cassetta, minidisco, video monitor. Programmazione in BASIC o in Assembler;
Challenger 3-OEM, elaboratore di elevatissime prestazioni (ha tre microprocessori: 6502, 6800 e Z80), con configurazione base di 32 Kb RAM e due unità floppy. I linguaggi di programmazione sono BASIC, Fortran, Cobol, Assembler;
Challenger 3-B, elaboratore con tre microprocessori (6502, 6800 e Z80), 48 Kbytes di RAM, due unità floppy ed una unità a disco (tecnologia Winchester) da 74 Mbytes. Questa configurazione può essere ampliata notevolmente.

Harden s.p.a. — **Commodore Business Machines, Inc.**, **Compucorp**
Via Giuseppina, 110
26048 Sospiro (CR)
Tel. (0372) 63137/63136
Telex 320588

Modello: PET 3032. Caratteristiche: 32 K RAM utente, CPU 6502, sistema operativo residente, Business BASIC, tastiera alfanumerica grafica (64 simboli grafici), video a 1000 caratteri (25 righe x 40 colonne). Periferiche: unità a cassette, stampanti, minifloppy disk.
Modello: Compucorp 655/665. Memoria RAM utente minima 24 K, max 36 K. Tastiera alfabetica + Pad numerico + 20 tasti definibili tre volte. Video da 12 pollici con 1600 caratteri (20 righe x 80 colonne). Periferiche: fino 4 minifloppy o fino a 2 dischi rigidi con supporto rimovibile (capacità 9,6 Mb ciascuno). Unità a nastro magnetico 7" IBM compatibile, 9 tracce. Stampanti: seriali o parallele.

Homic s.r.l. — **Commodore Business, Machines, Inc.**, **Hewlett-Packard Corp.**, **Nascom Microcomputers**, **Southwest Technical Products Corp.**
Piazza De Angeli, 1
20100 Milano
Tel. (02) 4695467

Prodotti Hewlett-Packard: calcolatrici programmabili ed il personal computer HP-85. Presentato dalla H.P. in gennaio, l'HP-85 è un personal computer integrato fornito di tastiera (72 tasti, di cui 4 dedicati a funzioni definibili dall'utente e 10 dedicati al controllo del cursore video e delle periferiche), cassetta magnetica digitale, stampante termica alfanumerica bidirezionale (32 colonne), video (16 righe x 32 colonne oppure 192 x 256 punti per la grafica). Memoria RAM di 16 K espandibile fino a 32 K. Linguaggio di programmazione: BASIC H.P. esteso.
Nascom 1: tastiera alfanumerica con 48 tasti, CPU Z80, memoria RAM 2 K, interfaccia video, interfaccia per registratore a cassette, interfaccia seriale RS 232, monitor da 1 K EPROM. Viene fornito in kit e/o montato. Sono disponibili schede per espansione memoria RAM, monitor da 2 K, tiny BASIC da 2 K o 3 K su EPROM.
Della Southwest Technical Products Corp è il sistema SWTPC 6809. Basato sul microprocessore 6809, ha un'espansione max di memoria RAM fino a 128 Kbytes, con possibilità di gestire più terminali. Memorie di massa: minifloppy, floppy da 8", disco rigido da 16 Mbytes. Interprete BASIC da 15 K.

Iret s.r.l. — **Apple Computer, Inc.**
Via Emilia Santo Stefano, 32
42100 Reggio Emilia
Tel. (0522) 49674/38213
Telex 530173 IRETRE I

Microcomputer Apple II Plus versione europea. Configurazione RAM possibili: 16/32/48/64 kbytes. BASIC esteso Applesoft in ROM. Compilatore Apple Pascal. Interfacce disponibili: seriale, parallela, Centronics, per comunicazioni. Periferiche: minifloppy, tavoletta grafica.

Larir International s.p.a. — **Heath Company**
Viale Premuda, 38/A
20129 Milano
Tel. (02) 795762

Modello H8: microprocessore 8080, memoria RAM di 4 K espandibile fino a 64 K con moduli di 4/16 K, interfaccia seriale, parallela e per cassetta magnetica. Periferiche: video-tastiera, stampante, minifloppy. Linguaggi di programmazione: Assembler, BASIC.
Modello H11A: microcomputer utilizzante moduli della serie LSI-11 della DEC.
Modello WH89: microcomputer integrato basato sullo Z80 e dotato di video-tastiera e minifloppy.

Mactronics Italia — **Diablo Systems, Inc.**, **Integral Data Systems**, **Intertec Data Systems, Inc.**, **Micropolis Corp.**, **Teletype Corp.**
Viale Jenner 40/A
20159 Milano
Tel. (02) 6882141
Telex 332452

Prodotti Diablo: stampanti a margherita, con stampa e movimento carta bidirezionali, possibilità di impiego come printer-plotter, interfaccia compatibile Centronics.
Prodotti Integral Data Systems: stampante IDS-440, 96 caratteri maiuscoli e minuscoli stampabili, densità di car/pollice e linea/pollice variabile, autotest incorporato,

segue a pag. 69

continua da pag. 64

DISTRUBUTORE	FABBRICANTE

stampa fino a 132 car/riga, interfaccia seriale RS232 e parallela compatibile Centronics.
Prodotti Intertec Data Systems: Intertube II, video display a 12", 24 righe per 80 caratteri, 128 caratteri ASCII visualizzabili, simboli speciali, blinking, video inverso, campi protetti e non, sottolineatura, indirizzamento del cursore, tastiera 128 caratteri ASCII con Pad numerico, autodiagniostica, porta per stampante, interfaccia RS232.
Microcomputer Superbrain: microprocessore Z80, memoria RAM di 32 K espandibile fino a 64 K, video display a 12" incorporato (capacità di 1920 caratteri, set grafico ASCII minuscolo e maiuscolo + 11 caratteri grafici speciali), tastiera alfanumerica (128 caratteri ASCII, Pad numerico, tasti funzione), due unità minifloppy, disco rigido da 10 a 300 Mbytes, compatibilità S-100 ottenibile tramite adattatore. Sistema operativo: CP/M.
Prodotti Micropolis Corp: floppy disk drive e microdisk drive (diam. 8", utilizzante tecnologia Winchester, con o senza controller intelligente, capacità non formattata fino a 45 Mbytes).
Prodotti Teletype Corp: terminali serie 43 (modelli con sola stampante, tastiera-stampante, con o senza lettore-perforatore di nastro, interfaccia RS232 o curret loop 20/60 mA), serie KSR o RO.

Microlem s.a.s. — **E & L Instruments**
Via Monteverdi, 5
20131 Milano
Tel. (02) 2710465

Sistema didattico MMD-1 con microprocessore 8080 o Z80 (tramite scheda di adattamento). Nella configurazione base comprende: RAM 1/2 kbyte, monitor per la gestione della tastiera (ottale), dei LED e di un registratore a cassette; basetta per montaggi sperimentali; alimentatore +5/+12/-12 V. Espansioni: memoria RAM-PROM, interfaccia stampante, registratore, video, tastiera alfanumerica ASCII, BASIC da 8 K, outboards per esperimenti.

OK Machine & Tool Italia s.r.l. — **OK Macchine & Tool Co.**
Via Ravizza, 34/1
20100 Milano
Tel. (02) 483463
Telex 320419 BRUS I

Gamma di prodotti ed accessori per il settore hobbistico. Utensili ed attrezzature di laboratorio, macchine, sistemi ed utensili per produzioni in wire-wrap.

P.B.S. s.r.l. — **APF Electronics, Inc.**
Via V. Monti, 15
20123 Milano
Tel. (02) 6107363

Personal computer Pecos I: memoria RAM 16 K espandibile fino a 32 K; sistema operativo di 24 K in ROM; tastiera alfanumerica, video (40 caratteri x 16 righe) e cassette magnetiche (80 K formattata) incorporati. Espansioni: max 4 cassette e max 4 floppy disk (doppia faccia, singola densità). Linguaggio di programmazione Pecos (derivato dal Joss della Rand Corporation).

Plae — **Cromemco, Inc.**, **Synertek, Inc.**, **Televideo, Inc.**
Via Curtatone, 16
20098 S. Giuliano Mil. (MI)
Tel. (02) 9880147

Personal computer Delta 1: basato sulle schede Sinertek SYM1 (single board computer con CPU 6502) e KTM2 (scheda di interfaccia video-tastiera ASCII), è dotato di interfaccia per cassetta magnetica e per stampante seriale (RS232). Linguaggio di programmazione: BASIC.
Prodotti Televideo, Inc.: terminali video serie 912 e 920. Controllati da microprocessore, hanno uno schermo da 12", 24 righe x 80 caratteri, intensità campi diversificata per presentazioni dati, possibilità del reverse mode, cursore indirizzabile, blinking, tastiera numerica, tasti funzione, editing, interfaccia RS232, uscita per stampante.

Radio Shack Italia s.r.l. — **Tandy Radio Shack Corp.**
Corso Europa, 12
20100 Milano
Tel. (02) 702406

Microprocessore Z80. Tastiera professionale integrata 53 tasti più tastiera numerica. Display: memory mapped; testo 16 righe di 64 caratteri con il BASIC level 1 e l'opzione di 32 caratteri per linea con il BASIC level 2. Grafici: 128 orizzontali per 48 verticali. Memoria centrale: modulare con capacità totale di 60 K. Sistema a minidisk: interfaccia di controllo a 4 drives. Stampante bidirezionale: matrice ad aghi 9 x 7; velocità di stampa 120 caratteri al secondo e 50 linee al minuto. Set grafico di 96 oppure 160 caratteri, sia maiuscoli che miniscoli (selezionabili).

Segi s.r.l. — **Epson America, Inc.**, **Hazeltine Corp.**
Via Timavo, 12
20124 Milano
Tel. (02) 6073255/6073088

Il miglior rapporto prezzo-prestazioni unito ad una incredibile semplicità costruttiva. I terminali Hazeltine comprendono una linea completa, dalla semplice teletype al terminale governato da due microprocessori. Interni; tutti hanno in comune l'affidabilità e la semplicità d'uso.

DISTRUBUTORE	FABBRICANTE

Superdattilo s.r.l. — **Lanier Business Products, Inc.**
Via Mantica, 28
33100 Udine

Il Word Processing System della Lanier Business Products, Inc. permette vari tipi di attività, quali statistiche, list/merge, word processing, etc. È possibile dotare il sistema di vari tipi di memoria di massa (floppy disk, magnetic disk cartridge, hard disk) o di stampanti, nonché collegare varie stazioni di lavoro.

Technitron s.r.l. — **Okidata Corp.**, **Pro-Log Corp.**, **Teleray Research**
Via California, 12
20144 Milano
Tel. (02) 4989279
Telex 332252 TECM I

Prodotti Okidata: stampanti economiche a 80 c.p.s. monodirezionali su carta normale, da 40/80/132 colonne, set di caratteri ASCII con 64 caratteri grafici.
Prodotti Pro-log: PROM-programmers Serie 90. Strumenti basati su microprocessori per programmare e testare PROM. Esistono 4 tipi di unità: M900 e M900B per progettazione e assistenza, M910 per produzione, M920 per l'utilizzo con TTY o CRT.
System analizers: strumenti di test per il monitoraggio delle operazioni di un sistema a microprocessore. Consentono l'analisi passo passo o in run e la sincronizzazione con oscilloscopio.
STD Bus microprocessor cards: l'STD Bus è un nuovo standard per microprocessori a 8 bits. La serie di schede comprende processori (Z80, 8085, 6800), memorie, I/O digitale TTL, I/O industriale.
Prodotti Teleray: terminale video semiintelligente controllato da microprocessore 6502, che consente fra l'altro operazioni di editing completo a controllo indirizzamento cursore. Disponibile anche con compatibilità a controllo indirizzamento cursore. Disponibile anche con compatibilità VT52, APL, DASHER (Data General).

Telcom s.r.l. — **Base 2**, **Microcomputer Power**, **Shugart Ass.**, **Summagraphics Corp.**, **Techtran Industries, Inc.**, **Trendcom**
Via Matteo Civitali, 75
20148 Milano
Tel. (02) 4047648
Telex 335654 TELCOM I

Prodotti Base 2: stampanti a impatto, 80 colonne.
Prodotti Microcomputer Power: alimentatori.
Prodotti Shugart Ass.: floppy disk serie 800, compatibile IBM, a singola e doppia densità; floppy disk serie 850, a doppia testa, singola e doppia densità, con capacità fino a 1,6 Mbyte; minifloppy SA400; minifloppy SA450 a doppia testa; unità a disco fisso, tecnologia Winchester, serie 1000 (8").
Prodotti Summagraphics Corp.: digitalizzatori BIT Pad a basso costo, adatto per applicazioni tradizionali, menu, console o tastiera di grande flessibilità e capacità operative.
Prodotti Techtran: unità a cassette magnetiche e a floppy disk munite di interfaccia seriale, costituenti terminali di emissione-ricezione dati o memorie esterne per teminali video e stampanti.
Prodotti Trendcom: stampanti a carta termica a 40 e 80 colonne, velocità 40 car/sec, prezzo contenuto, particolarmente adatte per collegamento a hobby, personal computers e terminali video.

Texas Instruments Semiconduttori Italia s.p.a.
02015 Cittaducale (Rieti)
Tel. (0746) 69034

Unicomp s.r.l. — **Cromemco International, Inc.**, **Exidiy Incorporated**
Via Cantù, 20
20092 Cinisello B. (MI)
Tel. (02) 6121041
Telex 331403 GEICO I

Gromemco Sistema 3: basato su microprocessore Z80, memorie da 64 a 512 kb in banchi da 64 K, due o quattro unità a dischetti da 512 kb ciascuno, collegabilità a uno o due dischi fissi da IOMB, da uno a sette video terminali, vasta scelta di stampanti. Linguaggi di programmazione: BASIC, Cobol, Fortran.
Personal Computer Sorcerer: collegato al televisore di casa e ad un normale registratore a cassette, permette di realizzare un economico e potente sistema di calcolo per l'hobbista e anche per il professionista. Unità centrale in Z80. Memorie da 8 a 48 K, più 4 K di monitor. Programmabile in BASIC (disponibile su ROM). Disponibile su ROM anche un potente word processor. Tastiera completa, con anche 128 caratteri grafici programmabili. Gestisce su video 1920 caratteri. Si può espandere per divenire un sistema completo con l'aggiunta dell'unità videodisk, che comprende due minidischi da 315 K ciascuno. Collegabile a varie stampanti.

Vianello s.p.a. — **Paratronics, Inc.**, **Systron Donner Corp.**
Via T. da Cazzaniga, 9/6
20121 Milano
Tel. (02) 3452071
Telex 310123 VIANE I

Analizzatore intelligente di stati logici; analizzatore logico nel dominio dei dati e del tempo con canale analogico per analisi di forme d'onda; programmatore di pilotaggio ed elaborazione per sistemi IEEE; calibratore campione volt c.c. per sistemi, programmabile IEEE; multimetro digitale per sistemi, programmabile IEEE.

PROGRAMMA CONFERENZE

A distanza di un anno l'elaborazione personale ripropone con BIT 80 il momento più qualificato e più specifico per l'esame di quanto in Italia si stà verficando.
Se da una parte già si parla di "seconda generazione" del personal computer, dall'altra il tipo di utenza ha assunto connotazioni ben circostanziate che vanno da coloro che vedono nei microcomputer un utile strumento per applicazioni hobbistiche/didattiche e chi invece identifica in questi sistemi "vestiti" più riccamente la soluzione a problemi di natura profesionale.
Per commentare i vari aspetti di questo settore, nel corso della rassegna si svolgerà il convegno:

L'ELABORAZIONE PERSONALE IN ITALIA

Convegno Coordinato dal Gruppo Editoriale Jackson

GIOVEDI' - 8 - MAGGIO - ore 10
Ing. M. Montedoro - Coordinatore della rivista BIT
Presentazione del convegno.

GIOVEDI' - 8 - MAGGIO - ore 10,30
SGS - ATES
Conversazione con il pubblico sul sistema didattico Nanocomputer.

GIOVEDI' - 8 - MAGGIO - ore 15
Dott. Occhini
Honeywell Information System Italia
Il personal computing come settore industriale.

VENERDI' - 9 - MAGGIO - ore 10
Dott. F. Fumi
Con le tue mani. Gioie e problemi del personal computer fatto da sé.

SABATO - 10 - MAGGIO - ore 10
Ing. Becattini e Ing. Pirri - General Processor
Real-T: un sistema multiprocessor per applcazioni gestionali-scientifiche.

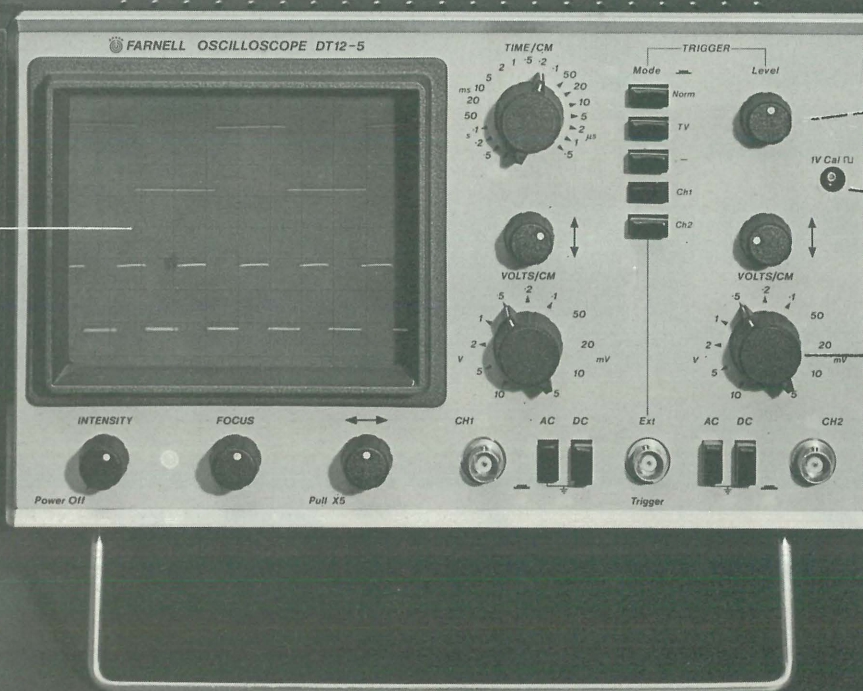
SABATO - 10 - MAGGIO - ore 15
Ing. Focardi - Unicomp
Motivazioni tecniche e di mantenibilità nelle scelte di prodotto della Unicomp.

DOMENICA - 11- MAGGIO - ore 10
Softec s.r.l.
La diffusione della microinformatica quale elemento nuovo della cultura contemporanea.

DOMENICA - 11- MAGGIO - ore 11
Walter Polk - Business Information Management Applic.
Le due esigenze fondamentali di un marketing information system efficace per il marketing: facilità di assimilazione del significato dei dati e "portabilità" del sistema.

DOMENICA - 11 - MAGGIO - ore 15
Ing. Mauro Salvemini - Università dell'Aquila
Computer grafica e grafica con il computer.

Siamo stati i primi . . .

a proporre un oscilloscopio professionale sotto il "Tetto" delle 500.000 lire. Ricordate il vecchio 12-4DA? è ancora il nostro più accanito concorrente: infatti chi l'ha acquistato (e sono stati in molti) non vuol saperne di cambiarlo. Ma guardate:

. . . Ora siamo gli unici

in grado di offrirvi una nuova generazione di oscilloscopi europei a doppia traccia, 12 MHz, ultracompatti (grazie al nuovo, ridottissimo, CRT che la Brimar ha sviluppato per noi) al prezzo di

486.000 LIRE

Farnell Italia s.r.l.

Via Mameli, 31 - 20129 MILANO - Tel. (02) 7380645 - 733178

DISPONIBILE A STOCK PRESSO:

SASSUOLO	- HELLIS	Tel. 059/804104
TORINO	- TOMEL	Tel. 011/743918
CHIAVARI	- GOLD	Tel. 0185/300773
ROMA	- SILV	Tel. 06/8313092
NAPOLI	- E.D.L.	Tel. 081/632335
BOLZANO	- RADIOMARKET	Tel. 0471/37407
TRIESTE	- RADIOKALIKA	Tel. 040/30341
CATANIA	- THYRISTOR	Tel. 095/444581
VERONA	- A.P.L.	Tel. 045/582633

Viste le caratteristiche, pensateci un po' non ne vale la pena?

☐ Desidero avere maggiori informazioni EK 5/80
☐ Desidero riservare il mio DT 12-5. Vogliatemi confermare le condizioni di acquisto.

Nome . . . .
Cognome . . . .
Ditta . . . .
Via . . . . N° . . . .
Città . . . . CAP . . . .
Tel. . . . .

*Prezzi validi al 29/2/80 - 1 lg = Lit. 1.780 IVA Esclusa Pag. alla consegna.

LA NOTA

Prime esperienze: un resoconto

di **Aldo Cavalcoli**

Nell'ultima nota apparsa sulla rivista BIT si diceva che "avevamo acquistato un Personal Computer". L'esperienza è andata avanti e ritengo senz'altro utile portare a conoscenza dei lettori alcune risultanze, soprattutto alcuni classici atteggiamenti psicologici nei confronti della macchina, chiarificatori di un certo approccio reale che viene a determinarsi, giusto o sbagliato che sia.

Prima cosa (orrore): i giochini.

Chiaramente dopo aver criticato e declassato i giochi, eccoci anche noi eccitati come non mai a giocare con il Personal, fino al punto,con la scusa di "imparare la macchina", di farci un misero Master Mind in italiano, quando qualsiasi buon libro sul BASIC riporta una valida versione di questo gioco universale.

Bene la macchina è appresa. E adesso?

Salta fuori il pallino della formatica.

E se si facesse un gioco istruttivo, che permette a ragazzi delle elementari di imparare qualcosa, tipo la zoologia?

Detto fatto ecco il programma "zoologico": dimmi il nome di un animale; è un mammifero? NO! È un uccello? SI! È un'aquila? ecc. ecc. Il bello è che cosa funziona, basata del resto su una interessante struttura ad albero, con la caratteristica che il programma impara, cioè, al crescere delle risposte esatte, cresce la sua capacità di dialogare con l'utente come, si spera, un futuro bambino della scuola elementare italiana.

Dopo i giochi, il gestionale più spinto, la fatturazione, prima quella personale (siamo o non siamo dei professionisti?), poi quella generale di una piccola azienda, con tutte le raffinatezze del caso.

Risultato: non ci sono limiti alla creatività ed alla soddisfazione.

Purtroppo il limite è subito imposto dalle rispettive mogli che iniziano a vedere nel Personal un gioco un po' troppo coinvolgente.

Per evitare la crisi famigliare è allora il caso di finalizzare l'niziativa e riportarla nel normale orario di lavoro, quindi, in altri termini, renderla "fatturabile".

Ma a chi? Ecco il grosso problema.

Individuare un'applicazione gestionale sufficientemente generale da raccogliere attorno a sé un numero di utenti minimo, tale da poter suddividere i costi in modo accettabile. Tutto è allo studio.

In ogni caso, l'esperienza è notevole, si tocca con mano che quando si dice e si vede in occasioni tipo il recente EDP USA è vero. Come il microprocessore è stato la causa di una rivoluzione nel mondo industriale, così il Personal lo sarà, se non lo è già, nel mondo del gestionale di piccole dimensioni, del privato da computerizzare, con le dovute cautele, naturalmente.

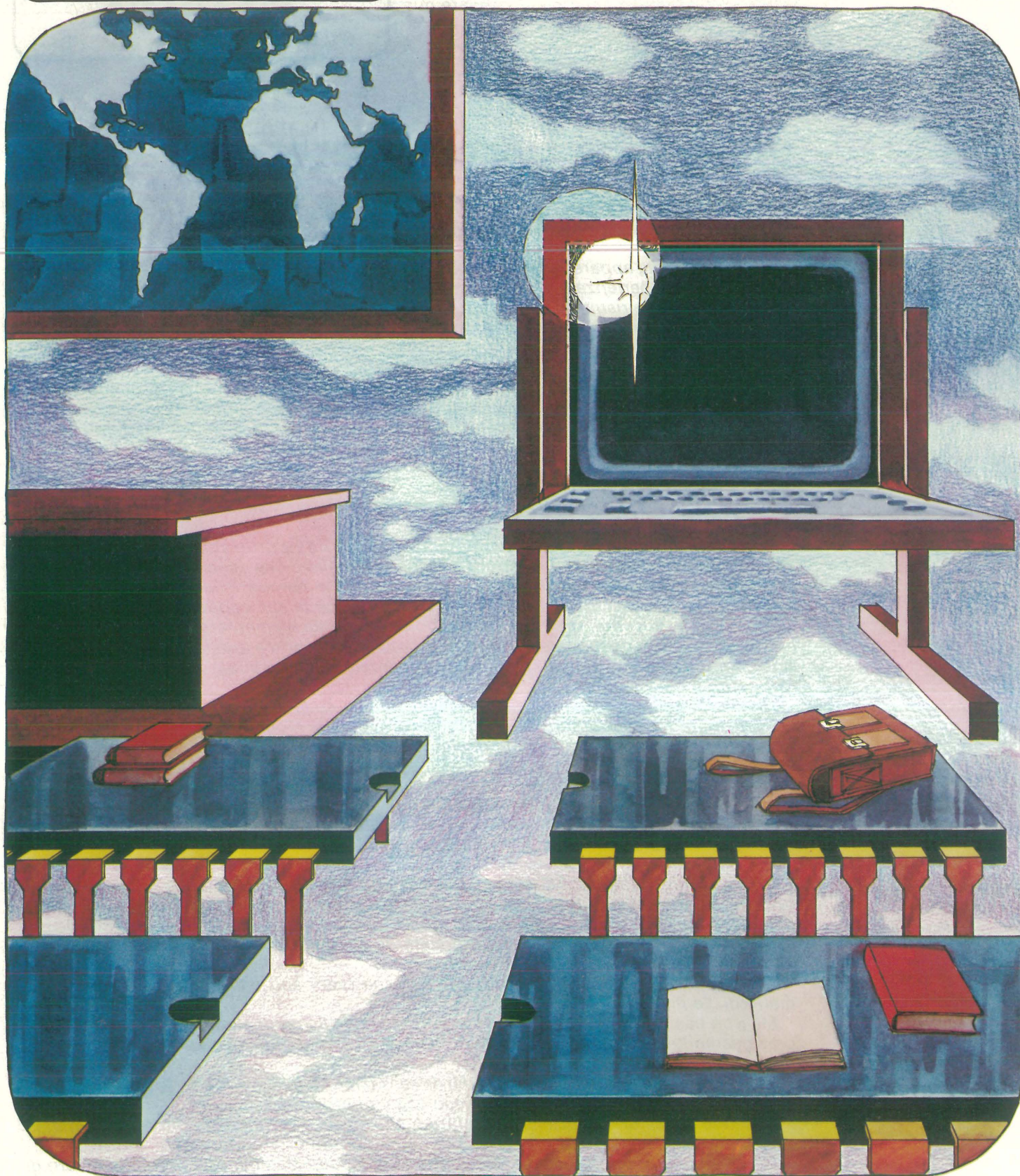
Occorre, come dice sempre il dottor Fontana (vedi suo editoriale sull'ultimo BIT), metterci "le mani sopra".

Ma non basta, occorre che l'utente si convinca che il Personal può fare molto per lui, e graziosamente esborsi le usuali esose parcelle all'esoso professionista che gli fa i programmi su misura.

Tornando al discorso dell'EDP USA 80, la sensazione è quella che è arrivato il momento: il Personal sta per esplodere come è esploso il microprocessore.

Ci siamo, e quindi, attenzione a non fare scelte errate o avventate. Molti si sono bruciati con i microprocessori: che lo stesso non capiti con un errato approccio ai Personal.

Un'idea: perchè non raccogliere esigenze simili soddisfacibili con un unico lavoro di progettazione? Siamo aperti ad ogni suggerimento e consiglio, scriveteci presso la redazione di BIT, sottoponendoci delle possibili applicazioni da voi fatte o pensate o desiderate.

Naturalmente daremo spazio solo a quelle più insensate.

Il calcolatore per i giovani
Il mondo dei numeri

di **F. Waldner** - *Università di Bari, Istituto di Fisica* ***Parte II***

La rappresentazione dei numeri interi e positivi

Nel numero precedente di BIT abbiamo parlato dell'informazione nel calcolatore, ed abbiamo visto che essa é esclusivamente ridotta nei suoi componenti elementari: i bits. Il raggruppamento in 8 bits (byte) è una scelta tecnica, che finora si è dimostrata vincente: ciò non toglie però che per il calcolatore il bit sia l'unica cosa "comprensibile".
Ne è conseguito che il sistema numerico ovvio per rappresentare dei numeri è il sistema binario, essendo il sistema esadecimale più o meno qualcosa che è comodo per noi, ma che è estraneo al calcolatore.
Vedremo in questa puntata come effettivamente si rappresentano i numeri in un calcolatore, che d'ora in avanti supporremo essere organizzato in bytes, nel senso che vedremo più avanti: per ora ci limiteremo a pensare che il byte è il più piccolo raggruppamento di bits che possiamo estrarre dal calcolatore, o che possiamo modificare *con una singola operazione*. Supporremo anche che i bytes nel calcolatore siano organizzati in parole di *due bytes*: quindi le parole risulteranno essere di 16 bits. Incidentalmente, questo tipo di struttura è molto diffusa e si sta diffondendo ancora di più: probabilmente sarà la configurazione comune a tutte le macchine del futuro.
Parliamo anzitutto dei numeri interi e positivi. Come potremo rappresentarli? Da quanto si è detto ciò dovrebbe risultare piuttosto chiaro: nel sistema binario, magari scrivendo un numero in un singolo byte. Qui però sorge subito una diffilcoltà: il byte ha 8 bits, ed il fatto che abbiamo parlato di numeri "positivi" ci fa presumere che in qualche modo (ancora da definire) dovremo dire alla macchina se stiamo parlando di numeri positivi o negativi. Poiché questa informazione si può esprimere con un bit solo, ne viene che ci resterebbero 7 bits a disposizione per poterci scrivere il nostro numero: ora qual é il numero più grande che possiamo scrivere usando 7 bits? Evidentemente è $(111\ 1111)_2$ che non è altro che il modesto 127 del sistema decimale.
Se usassimo quindi questa tecnica ci troveremo ben presto nei guai: anche numeri modesti come 475 *non* sarebbero rappresentabili nel calcolatore, il quale sarebbe obbligato, qualora se li trovasse davanti, magari come risultato di un'operazione, a dare una segnalazione di errore. Questa segnalazione di errore esiste in tutti i calcolatori, detto per inciso, e si chiama *'overflow'*, traduzione del nostro "trabboccare", proprio come l'acqua da una vasca.
Siamo quindi obbligati in pratica ad usare più bytes per rappresentare dei numeri interi: la scelta comune è di usare *due* bytes (cioé *una parola*) col che i numeri che si possono rappresentare diventano ragionevoli. Lascio a voi controllare che con 15 bits a disposizione potete rappresentare numeri fino a 32.767. Dopo di che avreste anche in questo caso, una segnalazione di overflow. Questa è la scelta comune, e di solito basta ed avanza per molte applicazioni pratiche: capita infatti molto di rado di dover usare numeri *interi* superiori a questo limite, almeno nel campo scientifico. Diverso può essere però il caso nel calcolo finanziario, di cui peraltro non parleremo in questi articoli. Quindi se si presentasse la neccessità di dover rappresentare numeri ancora più grossi, non avremmo altra scelta che quella di aumentare ancora i bits a disposizione: si usa infatti in questo caso una *doppia parola* (4 bytes) per rappresentare un numero, ed in tal caso il massimo numero rappresentabile diviene veramente grosso (oltre due miliardi, e per la precisione 2.147.483.653).
Si dice che in questo caso i numeri siano rappresentati *'in doppia precisione'* (double precision), ed evidentemente occorre stare attenti ad usare questa possibilità solo quando è strettamente necessario: infatti la quantità di memoria che viene così occupata è veramente rilevante. Non dimenticate infatti che in questo caso vengono *sempre* usati 4 bytes per rappresentare un numero, anche se questo è piccolo!
Possiamo riassumere quanto detto in una tabellina (v. Tabella I) in cui riportiamo i numeri massimi che si possono rappresentare con un dato numero di bytes, tolto un bit che servirà per sapere se un numero è positivo o negativo. Notiamo di passaggio che molto spesso *si sa* che i numeri che si useranno sono *piccoli* e *sempre positivi: se* questo è vero allora i programmatori furbi usano un solo byte (senza il bit del segno!) per rappresentare numeri di questo tipo. Il numero massimo che si può rappresentare in queste condizioni diventa 255.
Dobbiamo parlare ancora del *'bit del segno'*. Qui si tratta evidentemente di mettersi d'accordo: la scelta popolare è stata che il bit del segno è il bit più a sinistra, detto anche il bit "più significativo", nel senso che è legato alla più alta potenza di 2 (in inglese "*M*ost *S*ignificant *B*it", o *MSB* per brevità). Se questo bit è *zero* siamo in presenza di un numero positivo, se è uno siamo in presenza di un numero negativo.
Dire però che questa è stata una scelta popolare non è del tutto corretto; è stata anzitutto una scelta dettata da esigenze tecniche, come vedremo nel paragrafo successivo. Quella che invece è proprio una scelta "di comodità", è il criterio di numerazione dei bits, che vengono numerati in modo strano: a partire da zero (*non* da uno!) e a partire da destra (*non* da sinistra!). In figura 1 potete vedere com'é la situazione riferita ad una parola (cioé a 2 bytes).
Il bit 0 si chiama anche *LSB* come acrostico di "*L*east *S*ignificant *B*it", cioé "bit meno significativo".
Una parola va spesa per spiegare questo strano sistema di numerare i bits. Non è che sia una scelta del tipo "più complicato è meglio è": se ci pensate un momento vedrete che il numero del bit rappresenta la potenza di due a cui quel bit si riferisce. E questa

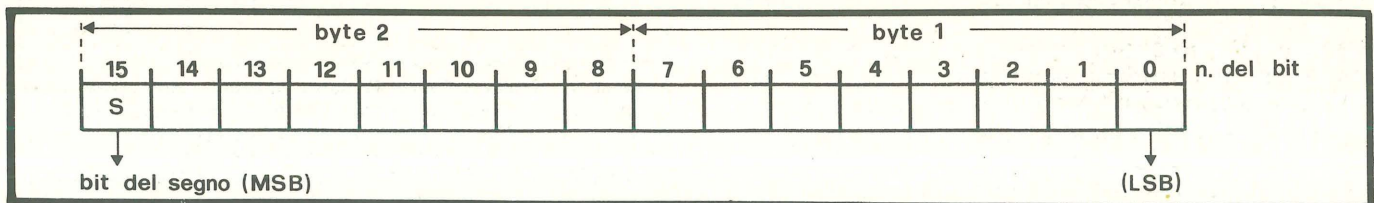



Figura 1 - Convenzione per la numerazione dei bits in una parola (2 bytes).

Numero di bytes	Numero massimo (decimale)	Numero massimo (esadecimale)
1	127 (255)	7F (FF)
2	32.767	7FFF
4	2.147.483.653	7F FF FF FF

Tabella I - Il numero massimo che si può rappresentare con uno o più bytes si ottiene mettendo ad 1 tutti i bits, tranne il bit più a sinistra, che è utilizzato come segno.

dovrete riconoscere che è una scelta molto comoda, che giustifica in pieno l'innaturalità di questo sistema di numerazione.
Con il che abbiamo detto praticamente tutto sul come si rappresentano i numeri interi e positivi all'interno di un calcolatore. È ora adesso di esaminare come si rappresentano i numeri interi e negativi, e ciò richiederà un discorso più ampio.

La rappresentazione dei numeri interi e negativi

C'era una volta un Re che aveva un Saggio Tesoriere, versato nelle Scienze Matematiche e nei Misteri dei Numeri.
Un giorno il Tesoriere cominciò a pensare alle profonde parentele che univano le operazioni dell'aritmetica ed a speculare sul fatto che forse tutte le operazioni potessero venire condensate e ricondotte ad *un unico tipo*. Il Tesoriere rimase particolarmente colpito dal fatto che addizione e sottrazione presentavano analogie sorprendenti: per esempio, se si sottraeva due da otto si otteneva (evidentemente) *sei*. Ma se si sommava ad otto la differenza fra dieci e due (che è otto) si otteneva sedici, e bastava togliere dieci da sedici per ottenere di nuovo sei. Il Tesoriere rimase impressionato dal fatto e chiamò la differenza fra dieci e due il *complemento a dieci* del numero due (quel numero cioé che aggiunto a due dà per risultato dieci), e si accorse di un fatto: invece di fare una sottrazione si poteva fare la somma del numero di partenza (minuendo) e del complemento a dieci del sottraendo: si otteneva un numero che differiva di *dieci* dalla differenza cercata.
Ciò non è un mistero, per noi. Siccome conosciamo il formalismo algebrico, possiamo fare molto più in fretta del Tesoriere: se ci ricordiamo che sommando e sottraendo uno stesso numero il risultato non cambia, potremo scrivere, per esempio:

$$7 - 4 = 7 + 10 - 10 - 4 \qquad (2.1)$$

e siccome il risultato di una somma non dipende dall'ordine degli addendi, il secondo termine dell'uguaglianza diverrà:

$$7 - 4 = 7 + (10 - 4) - 10 \qquad (2.2)$$

e poiché (10 - 4) è proprio il complemento a 10 di 4, vedete che il Tesoriere aveva proprio ragione. Non solo, ma al posto di 7 e 4 potreste scrivere un'altra coppia di numeri qualunque, e la proprietà or ora enunciata rimarebbe vera.
A noi tutto ciò interessa per il fatto che quando l'algebra booleana fu ben sviluppata, e si cominciò a voler lavorare con dei numeri (cioè effettuare in concreto delle operazioni) ci si accorse che il calcolatore, dovendo lavorare su bits, era in pratica in grado di fare *una sola* operazione aritmetica: la somma. *Tutte* le altre operazioni dovevano venir ricondotte in qualche modo a questa operazione fondamentale. *E così è ancora oggi.*
Non entriamo nel merito di come si riesce a fare moltiplicazioni e divisioni (è complicato, ma ci si riesce). Ci interessa invece sapere come si sottraggono due numeri: e la linea di ragionamento è la seguente:
— se a e b sono due numeri interi e positivi, scrivere a-b vuol dire *sommare* al numero positivo *a* il numero negativo *-b*. Fin qui nulla di nuovo rispetto all'algebra che tutti conoscono;
— visto che il nostro sistema di numerazione è in base *due*, la ricetta del Tesoriere ci dice che potremmo anche sommare ad *a il complemento a due di b;*
— allora è inutile che ci sforziamo di inventarci una rappresentazione strana per i numeri negativi: possiamo usare a questo scopo il complemento a due di un numero, ed abbiamo ciò che vogliamo;
— il risultato di tale somma non sarà però la differenza cercata: avremo un riporto di cui dovremo sbarazzarci, cosa del resto non certo difficile.
E questo chiude il discorso per numeri costituiti *da una sola cifra.* Una tabellina (v. Tabella II) vi permetterà di vedere con più chiarezza la situazione.
Resta da vedere come si può estendere il discorso a numeri *di più cifre*: basta osservare che il formalismo si può estendere, per cui è possibile scrivere

$$\begin{aligned} 35 - 13 &= 35 + 100 - 100 - 13 = \\ &= 35 + (100 - 13) - 100 \qquad (2.3) \end{aligned}$$

vedendo così che dobbiamo usare il *complemento a 100*, invece che a 10 per ritrovarci nel discorso testé fatto. Ma non è finita: potremmo anche scrivere

$$\begin{aligned} 35 - 13 &= 35 + 10000 - 10000 - 13 = \\ &= 35 + (10000 - 13) - 10000 \qquad (2.4) \\ &= 35 + 9987 - 10000 \\ &= 10022 - 10000 = 22 \end{aligned}$$

Se però supponiamo di avere solo quattro cifre a disposizione per rappresentare i numeri con cui abbiamo a che fare (ciò non capita se usiamo carta e matita, ma capita quando usiamo il calcolatore), vediamo che ci sono due strade per passare da 10022 a 22: la prima è di sottrarre 10000, e la seconda è di prendere solo le quattro cifre più a destra eliminando

M	S	M-S	S'	M+S'
0	0	0	10	[1] 0
1	0	1	10	[1] 1
1	1	0	1	[1] 0

Tabella II - Nella colonna M sono riportati i *M*inuendi, nella colonna S i *S*ottraendi, e nella colonna M-S le differenze risultanti. Nella colonna S' sono invece riportati i complementi a due dei sottraendi, e nella colonna M+S' sono riportate le somme dei minuendi e dei complementi a due dei sottraendi. Vedete che, se si pensa di togliere le cifre chiuse in riquadro, si ottengono proprio i numeri della colonna M-S.

puramente e semplicemente la cifra più significativa (*MSD* per *M*ost *S*ignificant *D*igit) dalle nostre considerazioni. E, detto per inciso, non è a caso che avendo 4 cifre a disposizione abbiamo scelto proprio 10000 come numero su cui effettuare il complemento: infatti $10000 = 10^4$.
Tutto questo è evidentemente immediatamente traducibile nel sistema binario. Se abbiamo numeri di 16 bits e dobbiamo effettuare una sottrazione, basterà che *sommiamo* al primo il complemento a 2^{16} del secondo e che ignoriamo il bit più significativo, (MSB), che comunque ci indicherebbe una condizione di errore; stiamo infatti tentando di andare *oltre* il numero più grosso, cioé siamo in *overflow:* ebbene, basta che in questo caso *ignoriamo la segnalazione di errore* per avere il risultato corretto della sottrazione.
E fin qui il discorso fila liscio. C'è solamente un intoppo: occorre avere un metodo pratico e semplice per ottenere il complemento del sottraendo. Se questo metodo si dovesse rivelare troppo complesso, tutta l'eredità spirituale del Tesoriere sarebbe praticamente inutilizzabile.
Esaminiamo allora come possiamo ottenere rapidamente il complemento a 10000 di 13, che scriveremo 0013. Basterà fare il *complemento a 9* delle prime tre cifre, ed il *complemento a 10* della sola ultima cifra: otteremo, appunto, 9987.
Oppure potremmo seguire un'altra strada: fare il complemento a 9 di *tutte le cifre* (ed otteremo 9986) e sommare 1 alla fine (ed otteremo di nuovo 9987).
Possiamo tradurre tutto ciò nel sistema binario: per ottenere il complemento a due di un numero scritto in codice binario dovremo seguire i seguenti passi:
— scrivere il complemento a 1 del numero stesso
— sommargli 1
Abbiamo spostato il problema: ora dobbiamo scoprire come si fa praticamente a scrivere il *complemento a 1*. Ed allora sarà meglio che vediamo i vari casi che ci si possono presentare, i quali - per nostra fortuna - non sono molti:
— il complemento a 1 del numero 0 è 1 (dobbiamo sommare a 0 proprio 1 se vogliamo ottenere 1 come risultato)
— il complemento a 1 di 1 è 0
— quindi, se abbiamo un numero scritto in codice binario (come una sfilza di 0 e 1) potremo scrivere *subito* il suo "complemento ad 1" *se cambieremo tutti gli 1 in 0 e viceversa.*
E spero riconoscerete che questa operazione è estremamente semplice da eseguire, anche per quella macchina estremamente stupida che è un calcolatore. Se poi vogliamo il *complemento a due* non dovremo fare altro che aggiungere 1 al risultato ottenuto. Ed è tutto.
Con questa ricetta possiamo dire che i numeri negativi nel calcolatore verranno rappresentati (se siamo in presenza di numeri di 16 bits) come "complementi di 2^{16}": si dice anche, per brevità, come "complementi a 2". *In questa rappresentazione somme e sottrazioni ri ridurranno a sole somme,* secondo la ricetta del Tesoriere.

Qualche esempio pratico

Supponiamo di voler verificare che 13 - 13 = 0. Il numero 13 si scrive:

0000 0000 0000 1101 in binario

oppure

00 0D in esadecimale.

Il complemento a 1 (occorre cambiare tutti i bits) diviene:

1111 1111 1111 0010 (FF F2 in esadecimale)

Sommando 1 a questo, otteniamo il complemento a 2:

1111 1111 1111 0011 (FF F3 in esadecimale)[1]

Se sommiamo ora 13 ed il suo complemento a 2 avremo

$$\begin{array}{rrrrr} & 0000 & 0000 & 0000 & 1101\ + \\ & 1111 & 1111 & 1111 & 0011\ = \\ \hline 1 & 0000 & 0000 & 0000 & 0000 \end{array} \qquad (2.5)$$

che è proprio zero, se cacciamo via il MSB, come da ricetta.
Da questo esempio vediamo anche che il bit 15 è zero per numeri positivi, ed è 1 per numeri negativi (scritti nella loro forma del complemento a 2).
Altro esempio: 135 - 73 = 62. Sinteticamente abbiamo:

$$\begin{array}{l} (135)_{10} = (0000\ \ 0000\ \ 1000\ \ 0111)_2 = (00\ 87)_{16} \\ (73)_{10} = (0000\ \ 0000\ \ 0100\ \ 1001)_2 = (00\ 49)_{16} \\ \quad (1111\ 1111\ 1011\ 0110)_2 = (FF\ D6)_{16} \\ \quad \text{......complem. a 1 di 73} \\ \quad (1111\ 1111\ 1011\ 0111)_2 = (FF\ D7)_{16} \\ \quad \text{......complem. a 2 di 73} \end{array} \qquad (2.6)$$

e la nostra operazione diventa (lascio a voi la verifica):

$$\begin{array}{rrrrr} & 0000 & 0000 & 1000 & 0111\ + \\ & 1111 & 1111 & 1011 & 0111\ = \\ \hline 1 & 0000 & 0000 & 0011 & 1110 \end{array} \qquad (2.7)$$

Concludendo, quindi, possiamo dire che:
— nel calcolatore i numeri negativi vengono rappresentati nella loro forma di *complemento a due*: la ricetta per ottenerli è di cambiare tutti i bits,al numero di partenza e di sommare 1 alla fine
— il MSB è interpretabile come il *'bit del segno'*: vale 0 per i numeri interi e positivi e 1 per quelli negativi (cioé per i numeri scritti come *'complementi a due'*.
— somme e sottrazioni nel calcolatore si riducono alla sole somme fra numeri normali o rappresentati come complementi a due.
Adesso potrete provare a fare un po' di pratica: tentate di fare le seguenti somme e sottrazioni, sia in decimale, che in binario, che in HD. Nella prossima puntata vi darò (al solito) le soluzioni, che però stavolta saranno date - per brevità - nel solo HD. Ecco quindi le operazioni proposte:

65 - 128 = ...; 1234 + 735 = ...;
12856 - 31745 = ...; 17267 + 22456 = ...;
6547 - 17722 = ...; -1378 - 1255 = ...;

---

[1] *D'ora in poi scriveremo* HD *per* Hexa-Decimal*.*

Il programma in BASIC

La volta scorsa abbiamo visto come si costruisce un programma comprensibile alla macchina, a partire da un programma comprensibile a noi: nel BASIC - abbiamo concluso - tale operazione è affidata ad un programma standard che viene fornito in genere dalla stessa Casa che costruisce il calcolatore, e che si chiama *interpretatore*.
Non sarà male a questo punto che spendiamo due parole per dire *che cosa è* un programma. Intendiamo con questo termine una serie di operazioni che devono venir effettuate su dei dati; questi *possono* essere dei numeri, ma non *necessariamente*: il bit infatti - ricordate - è in grado di rappresentare qualunque tipo di informazione, non solamente quella numerica. Una volta specificata la serie delle operazioni, il calcolatore le eseguirà una per una *in sequenza*: e questo è tutto ciò che il calcolatore è in grado di fare, non essendo evidentemente in grado di pensare per conto suo, altro che nei romanzetti di fantascienza o nelle elocubrazioni di intervistatori e giornalisti.
Se però fosse solo questo ciò che il calcolatore può fare, il numero di programmi che si potrebbero scrivere sarebbe piuttosto limitato. Pensate infatti ad una semplice equazione di 2° grado del tipo solito:

$$a\,x^2 + b\,x + c = 0$$

È evidentemente possibile dire al calcolatore ciò che deve fare per ottenere le soluzioni dell'equazione, noti che siano i coefficenti a, b, c: basta infatti tradurre in istruzioni la ben nota formula di soluzione. C'é però un intoppo: nel caso in cui il discriminante dell'equazione sia minore di zero la formula non funzionerebbe. Occorre quindi da un lato che venga previsto dal programmatore anche questo caso, e dall'altro occorre che il calcolatore sia capace di accorgersi di questa condizione, e di prendere una decisione opportuna: occorrono quindi nell'insieme delle istruzioni del calcolatore anche delle istruzioni del tipo "se succede questo fai così, se succede quello fai cosà". Tali istruzioni prendono il nome di "istruzioni logiche".
Quindi *tutto* ciò che il calcolatore deve fare, *tutte* le strade che potrebbe prendere devono venir previste e descritte dal programmatore: indubbiamente quando poi il calcolatore eseguirà il programma *sembrerà che pensi*. In realtà si limiterà ad eseguire diligentemente il pensiero di chi ha programmato. Ritenere che esso "pensi" non è molto diverso dal ritenere che una fabbrica "costruisca" automobili o che sia un cantiere edile a "far crescere" una casa.
Per orientarsi di fronte a tutte le strade che si possono presentare in un programma si usa di solito un sistema grafico che prende il nome di *diagramma di flusso* o di *flow-chart*. Non badate a chi vi dirà che è una cosa difficile, complessa, e vi confonderà le idee con un mare di simboli grafici. In realtà un eccellente diagramma di flusso può venir costruito con due soli simboli, entro cui si scrivono a parole le operazioni da effettuare:

- il *rettangolo*, per le operazioni da effettuare sui dati
- il *rombo*, detto anche "diamond" (diamante) in inglese, per indicare delle decisioni logiche.

Con questa semplice simbologia il nostro programma delle equazioni di 2° grado verrebbe descritto da un disegno come quello di figura 2, che credo a questo punto si spieghi da sé.

L'ambiente del BASIC

Anche questo infatti ha la sua importanza. Supporremo di essere in presenza di un calcolatore-tipo, quale oggi può essere un medio calcolatore da tavolo, o un personal computer più o meno da sogno. Gli attrezzi necessari sono in genere i seguenti:

- il calcolatore: *CPU*, o *C*entral *P*rocessing *U*nit con annessa memoria.
  Questa è in genere la parte che costa meno, il vero cuore del sistema;
- un display grafico, in genere un tubo a raggi catodici (d'ora in avanti *VT* per *V*ideo *T*erminal);
- una tastiera tipo macchina da scrivere, con lettere, numeri e simboli perciò detta *alfanumerica* (*TTY* d'ora in avanti, per *T*ele *TY*pe);
- un disco, piccolo o grosso che sia (detto *DK*, per *D*is*K*, d'ora in poi).
  A questo proposito avrete sentito parlare anche di *floppy disks*: non sono altro che dischi flessibili e sottili ed a parte il fatto che la registrazione è magnetica, non si presentano in modo diverso da certi "dischi omaggio" inseriti in certi giornali. Per inciso "floppy" vuol dire proprio "floscio, molle";
- un nastro magnetico, magari una semplice musicassetta. D'ora in poi lo chiameremo *MT* per *M*agnetic *T*ape. Dischi e nastri servono per immagazzinare dati e programmi, come una "memoria lenta", se volete, ma molto grossa. Vengono anche detti - soprattutto i grossi dischi - *mass storage*, cioé *memoria di massa*;
- una stampante su carta, per far uscire la lista dei programmi ed i risultati in modo leggibile e permanente. D'ora in avanti verrà detta anche *LP* per *L*ine *P*rinter, cioé "stampante in linea" (in linea col calcolatore si intende);

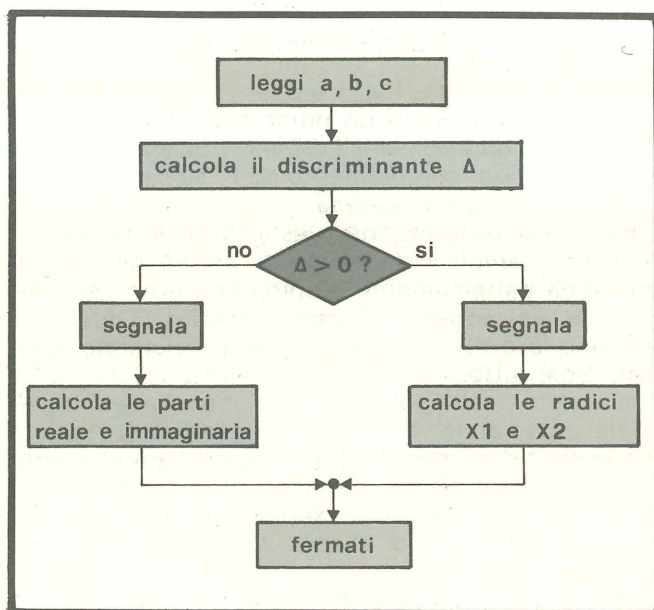



Figura 2 - Il calcolatore esegue le indicazioni che gli forniamo con il programma. Il diagramma di flusso è utile a noi per scrivere il programma, ci permette di analizzare le varie strade che ad esempio un calcolo può richiedere.

— una serie di programmi di gestione del calcolatore medesimo, fra cui l'interpreter del BASIC.

I punti precedenti tranne l'ultimo vengono detti anche *hardware*, con un antico termine mercantile che indicava letteralmente le "merci dure", cioé le "ferramenta". L'ultimo punto è chiamato per contrasto "merce molle", cioè *software*. E questi termini pittoreschi fanno parte ormai del linguaggio tecnico corrente.
Supponiamo quindi di aver acceso tutto il sistema, e di aver eseguito tutta una serie di operazioni che di solito sono specificate nei manuali del nostro calcolatore. Ci siederemo di fronte alla TTY (il VT è di solito posto sopra questa) e dopo un dialogo iniziale inteso a caricare l'interpreter del BASIC vedremo apparire la parola READY che ci informerà che la macchina è pronta a ricevere il nostro programma. Supporremo che questo sia proprio il programma per la risoluzione di un'equazione di 2° grado, di cui sopra.
Dovremo quindi battere a macchina una istruzione alla volta, facendola precedere da un numero progressivo, che servirà ad identificarla.
Poiché però dovremo prevedere anche correzioni ed inserzioni di altre istruzioni (nessuno è perfetto!), non sarà male che dopo l'istruzione 10 non scriviamo l'istruzione 11, ma magari la successiva istruzione la chiamiamo 15, per esempio.
È invalso l'uso di avere una numerazione progressiva, all'inizio, scalata di 10 in 10.
Scriveremo anzitutto *cosa* il programma fa: queste *non* sono istruzioni che il programma dovrà eseguire, ma un semplice pro-memoria per noi, e vengono dette quindi *istruzioni di commento* (comment statements). Si scrivono facendo seguire al numero dell'istruzione uno spazio (e ciò vale in tutta generalità per il BASIC: il numero dell'istruzione è *sempre* seguito da uno spazio) e quindi dalla parola REM (abbreviazione di REMARK, cioé commento). Dopo la quale si può scrivere ciò che si vuole: esso verrà stampato ogniqualvolta si chiederà la lista del programma, ma non avrà alcun effetto sull'elaborazione. Usando il BNF spiegato nel numero 5 di BIT, potremo sintetizzare il tutto così:

<istruzione di commento>::= < numero> *REM* <qualunque frase> (2.8)

e d'ora in avanti ometteremo anche di ricordarvi ogni volta di specificare il numero dell'istruzione, considerandolo ovvio e sottointeso. La (2.8) diverrà quindi

<istruzione di commento>::=*REM* <qualunque frase> (2.9)

Nel programma che stiamo costruendo potremo quindi scrivere i primi statements di commento nel modo seguente:

```
0010 REM __________
0020 REM PROGRAMMA PER CALCOLARE
0030 REM LE SOLUZIONI REALI O COMPLESSE
0040 REM DI UN'EQUAZIONE DI SECONDO GRADO
0050 REM DATI I COEFFICIENTI
0060 REM __________
```

È da notare una cosa: nel BASIC tutte le istruzioni vanno scritte *su una solo riga*, in altri termini *non si può* continuare un'istruzione nella riga successiva, ed il perché dovrebbe risultarvi evidente da quanto si è detto nel numero precedente circa la funzione dell'interpretatore, che traduce ed esegue il programma *una riga per volta*.
Però si possono scrivere *più* istruzioni su una stessa riga.

Costanti e variabili

Sarà bene chiarire che le *costanti* in un programma rappresentano dei dati che non possono in alcun modo venir fatti variare nel corso dell'esecuzione. Una *variabile* invece assume vari valori nel corso dell'esecuzione.
Nel BASIC le costanti sono solo numeriche ed i numeri si possono scrivere in tre modi diversi, che sono peraltro trattati dal calcolatore sempre nello stesso modo:
— come numeri interi
— in notazione decimale
— in notazione scientifica.
Supponiamo di voler scrivere il numero 1375. Nella notazione intera scriveremo - appunto - 1375, *senza punti o virgole*. Invece, se usassimo la notazione decimale, dovremmo scrivere o 11375. o 1375.0 o 1375.00, e così via. In altri termini è richiesta la *presenza del punto decimale*, seguito oppure no da ulteriori cifre.
Qui sorge un problema: di quante cifre può essere composto un numero? Qui purtroppo non vi posso dare una risposta, giacché questa dipende dal calcolatore che viene usato e dall'interpretatore che in esso sta lavorando. Normalmente dieci cifre bastano ed avanzano per tutte le operazioni usuali, ed è un errore pensare che dare dati con molte cifre faccia aumentare la precisione dei calcoli!
Potremo scrivere anche il nostro numero in notazione scientifica. Con questa frase si intende quella notazione numerica che separa il numero nel prodotto di una parte relativamente piccola (detta *mantissa*, altra parola araba!) e di una potenza di 10 (detta *ordine di grandezza*).
Chiariamo le cose con un esempio: il nostro numero 1375 può venir scritto nei seguente modi:

$$0.1375 \cdot 10^4;\ 1.375 \cdot 10^3;\ 13.75 \cdot 10^2;\ 137.5 \cdot 10^1 \text{ etc.} \quad (2.10)$$

Mentre però quando siamo noi a dare dei valori al calcolatore possiamo usare una qualunque delle rappresentazioni sopra elencate, il calcolatore ci fornirà dei dati solo nella prima forma, quella sottolineata, con la mantissa, cioé, compresa fra 0 e 1.
La notazione scientifica possiede evidenti vantaggi: può rappresentare in forma compatta e non equivocabile numeri positivi e negativi (la mantissa può essere anche negativa, evidentemente!), sia molto grandi (ed allora l'esponente di 10 sarà grande e positivo) che molto piccoli (ed allora l'esponente di 10 sarà grande e negativo).
È scelta comune che i numeri più lontani dallo 0 rappresentabili dal calcolatore siano $+10^{99}$ e -10^{99}, e quelli più vicini siano -10^{-99} e $+10^{-99}$: sono in realtà numeri enormemente grandi e fantasticamente piccoli!

Soluzioni degli esercizi presentati nel numero precedente (BIT N. 6)

Es. 1 Scrivete in binario e in decimale i seguenti numeri esadecimali:

1F, CA, FE, E2, 7C, 77, 99

R: $(1F)_{16} = (0001\ 1111)_2 = (31)_{10}$
$(CA)_{16} = (1100\ 1010)_2 = (202)_{10}$
$(FE)_{16} = (1111\ 1110)_2 \times (254)_{10}$
$(E2)_{16} = (1110\ 0010)_2 = (226)_{10}$
$(7C)_{16} = (0111\ 1100)_2 = (124)_{10}$
$(77)_{16} = (0111\ 0111)_2 = (119)_{10}$
$(99)_{16} = (1001\ 1001)_2 = (153)_{10}$

Es. 2: Convertite in esadecimale i seguenti numeri decimali:

7, 34, 92, 137, 232, 74, 143, 32

Potete inventare un metodo, analogo a quello che vi ho spiegato, per effettuare questo tipo di operazione?

R: $(7)_{10} = (7)_{16}$; $(34)_{10} = (22)_{16}$; $(92)_{10} = (5C)_{16}$; $(137)_{10} = (89)_{16}$; $(232)_{10} = (D8)_{16}$; $(74)_{10} = (4A)_{16}$; $(143)_{10} = (8F)_{16}$; $(32)_{10} = (20)_{16}$

Per quanto riguarda una regola pratica per effettuare la conversione, la cosa è molto più semplice: si può usare lo stesso formalismo spiegato nel numero scorso, avvalendosi di successive divisioni per 16: sotto ad ogni numero si scrive la sola parte intera del quoziente, mentre il resto lo si scrive nella colonnina di destra Per esempio:

```
1475   3
  92  12
   5
```

Infatti 1475 : 16 = 92 col resto di 3; 92 : 16 = 5 col resto di 12. A questo punto le cifre del numero in esadecimale vanno lette "alla rovescia", come al solito, e poiché $(12)_{10} = (C)_{16}$ avremo che il nostro 1475 (in base 10) si scrive 5C3 in esadecimale.

Es. 3: Provate (basandovi su ciò che sapete delle somme col sistema decimale) a definire le regole per sommare insieme numeri binari ed esadecimali. Per esempio

```
 10 1101 +        1AB32 +
  1 0110 =         724F =
 .........       ........
```

R: *Occorre anzitutto costruire una "tabella delle somme", che poi non è altro che l'analogo di ciò che le maestre ci hanno insegnato nelle elementari per il sistema decimale: ci è stata insegnata in effetti tutta una serie dei possibili casi che si possono presentare quando si sommano due cifre decimali: 2 + 2 = 4, però 6 + 6 = 12, ed allora "scrivo due e riporto 1" questo "1 di riporto" va aggiunto alla somma che si farà nella colonna di sinistra, e così via. Nulla cambia nei nuovi sistemi, a parte la tabella delle somme, che è bene sistemare in righe e colonne, mettendo all'incrocio di una riga e di una colonna il risultato, col riporto fra parentesi. Una volta capito il trucco potete farvi da voi la tabellina per il sistema esadecimale: qui vi riporto solo la tabellina del sistema binario, dato che l'altra ha dimensioni veramente proibitive (e vedrete quando comincerete a scriverla: vi consiglio di prendere un foglio piuttosto grande!). Detto per inciso: la tabellina delle somme, che costa tanta fatica ai bambini delle elementari, è una tabella 10 x 10, cioé di 100 numeri. Quella esadecimale è invece una tabella 16 x 16, cioé di ben 256 numeri. Siccome non è pensabile imparare a memoria una faccenda del genere, ci si limita di solito a convertire mentalmente gli addendi al sistema decimale, (cosa non difficile, trattandosi di numeri di una sola cifra), effettuare la somma nel sistema decimale, convertire la somma in esadecimale e quindi scrivere il risultato, con eventuali riporti. Così, nel caso della puntata precedente, dovendo fare 2 + F ci si ricorda che F è il 15 decimale, allora 2 + 15 = 17 (decimale), cioé 11 esadecimale, quindi "scrivo 1 e riporto 1". E così via.*
Eccovi i risultati delle operazioni proposte:

10 1101 + 1 0110 = 100 0011

(in decimale sarebbe 45 + 22 = 67).

1AB32 + 724F = 21D81

(in decimale sarebbe 109·362 + 29·263 = 138·625).

Ed eccovi la tabellina delle somme per il sistema binario

+	0	1
0	0	1
1	1	(1)1

Per completare il quadro, ecco un metodo rapidissimo per convertire un numero esadecimale in decimale, con l'aiuto di un calcolatore tascabile, di cui ognuno di voi dovrebbe essere tassativamente munito, sapendolo oltre tutto usare con disinvoltura. Meglio se scientifico (anzi, al giorno d'oggi sarebbe praticamente indispensabile) e meglio ancora se programmabile. Supponete di voler convertire 724 F: i passi da seguire sono i seguenti:

- *moltiplicate 7 per 16*
- *sommate 2*
- *moltiplicate per 16*
- *sommate 4*
- *moltiplicate per 16*
- *sommate F, cioé 15*

Questo è il metodo più pratico che io conosca per questo tipo di conversione.

Calcolo binario e logica. Il calcolo binario con le sue cifre 0 ed 1, proposto da Leibniz, e la logica binaria sono alla base della struttura degli elaboratori, ma, prima ancora di una razionalizzazione di questi principi, esistevano tecniche che ne anticipavano i concetti. Leibniz credette di vedere ne I Ching, antica tecnica divinatoria cinese, un uso primitivo del sistema numerale binario. Per la consultazione de "Il libro dei mutamenti" i cinesi facevano infatti ricorso a permutazioni (ottenere con estrazioni a sorte) a sei di due elementi: linea intera e linea spezzata. La tavola mostra I e 64 (2^6) permutazioni possibili.

Se mi permettete un inciso, vorrei spendere due parole per incoraggiare l'uso di questa notazione in potenze di 10, anche nel parlare comune, evitando orribili frasi tipo "centinaia di migliaia di miliardi" e simili piacevolezze care ai politici e ai presentatori TV. Forse per essere chiari alle volte bisogna essere un po' difficili all'inizio, ma almeno non si è totalmente incomprensibili in seguito!
Tornando al calcolatore, il nostro 1375 lo potremo scrivere 0.1375E+4 oppure 0.1375E4 dove E è l'abbreviazione di *E*xponent e il 4 è - evidentemente - proprio l'esponente di 10: in altri termini, si scrive solo la mantissa (positiva o negativa a seconda delle necessità) e l'esponente di 10 del numero. Ecco alcuni esempi pratici, al solito.

$$15745.86 = 0.1574586 \cdot 10^5 \qquad (2.11)$$
$$0.000756 = 0.756 \cdot 10^{-3}$$
$$-6547.78 = -0.654778 \cdot 10^4$$
$$(-0.0000006578 = -0.6578 = 10^{-6}$$

Questo è tutto ciò che c'è da dire su come si rappresentano le costanti. Quanto alle variabili la cosa è molto più semplice:
una variabile si può rappresentare con un simbolo costituito o solo da una lettera seguita da una singola cifra. Quindi nomi di variabili possono essere A, B, H, K5, V4, F9, etc.
E per ora questo è tutto. La prossima volta vedremo come si scrivono le espressioni matematiche e come si introducono i dati nel calcolatore. ■

Trasmissione seriale: concetti di base

Parte I di **B. Carbone**

Come collegare fra loro microcomputer e terminali.

Premessa

Eccoci qui, tra terminale e microcalcolatore, con due connettori in mano a chiederci cosa succederà quando li avremo collegati.
Esperienze acquisite nei nostri anni più verdi ci tornano precipitosamente alla memoria per ammonirci che collegare tra loro due "trabiccoli" non è più così elementare.Non bisogna tuttavia, in una situazione simile, farsi prendere dalla paura; basta sapere semplicemente cosa vuole un'apparecchiatura e cosa offre l'altra nella connessione, ed il gioco è fatto.
Parleremo qui di seguito di connessione tra un terminale (Teletype-TTY, terminale video-CRT, cassetta magnetica, ecc.) ed un microcalcolatore, e sono sicuro che moltissimi di voi mi seguiranno. Perché?

— Perché alcuni di voi, frequentando negozi e mostre, si saranno accorti che terminali stampanti e video, in effetti, non sono poi molti, anche se tanti hanno etichette diverse (magari con il marchio del micro a cui sono connessi e con il quale sono venduti). Un rapido confronto tra i preventivi che vi siete fatti rilasciare per ogni *singola* voce del sistema ha posto in evidenza che il terminale (lo stesso), nelle varie configurazioni, nei diversi sistemi, vien fatto pagare a volte a prezzi molto diversi. E allora perché pagare X+Y lire per il terminale "Tizio" (con etichetta "Caio") collegato al micro "Caio", quando lo stesso terminale venduto direttamente dalla casa "Tizio" che lo costruisce costa X lire? "Sono già risolti i problemi d'interfaccia — rispondono i venditori — è possibile collegarlo con tranquillità". Questa tranquillità però, il più delle volte, è fatta pagare un po' cara!

— Perchè comincia a comparire, per ora timidamente, ma sicuramente registrerà grossi sviluppi, il mercato dei terminals usati. Sono questi, terminali fatti mettere in pensione da grossi utenti EDP per le prestazioni ritenute superate (per esempio il baut-rate). Niente di meglio per l'hobbista con minori pretese che attingere a questo mercato per soddisfare le proprie esigenze con risparmi davvero rilevanti.

— Perché, crescendo la nostra esperienza, prima o poi ci facciamo cogliere dalla tentazione di ampliare il nostro sistema e di collegarlo a terminali aggiuntivi alla usuale console.

Quanto segue ha l'intento di fugare i vostri timori nell'accettare la decisione di effettuare da sè i collegamenti in situazioni simili, fornendo indicazioni e suggerimenti d'ordine pratico.

Un approccio elementare

Cercherò qui di seguito di chiarire alcuni concetti fondamentali di comunicazione dati facendo ricorso ad un "esempio". I più esperti non me ne vogliano se il tutto può apparire loro banale.

— Seriale/Parallelo

Cominciamo, nell'intento di semplificare il discorso, col parlare di comunicazione tra una macchina che genera messaggi (il microcalcolatore, ad esempio, oppure noi su una tastiera) ed un'altra che li riceve (ad esempio un terminale stampante). Sappiamo che il messaggio è costituito da un insieme di unità elementari di informazione: i *bits*.
Gruppi di bits (ad esempio 7) costituiscono un *carattere*.
Quando in un collegamento è possibile trasmettere un carattere intero per volta nell'unità di tempo, si parla di collegamento *in parallelo*. Occorrono evidentemente, nel caso di caratteri a 7 bits, 7 fili diversi uno per ciascun bit del carattere da trasmettere.
Quando invece si dispone di un solo filo per la trasmissione del messaggio, la trasmissione stessa può avvenire solo in modo *seriale*. I caratteri cioé sono trasmessi in linea un bit alla volta in intervalli di tempo successivi.
Per spiegarvi ancor meglio la cosa ricorro ad una esemplificazione "visiva".
Immaginate di avere tante palline con un foro che le attraversi da parte a parte. Supponete che siano dipinte di bianco oppure di nero: questi sono i bits.
Supponete anche di disporre di bastoncini su cui potete infilare e fissare più palline assieme, ad esempio 7: questi, con le palline relative sono i caratteri.
Supponete ancora di avere un amico che abiti nella casa di fronte, ad un piano più basso rispetto a voi, a cui volete fare arrivare un messaggio (un insieme numeroso di palline). Se disponete di un piano inclinato da disporre tra voi e lui, il gioco è fatto. Questo piano è il cavo di collegamento. Se il piano è abbastanza largo da lasciar scivolare un bastoncino intero per volta, nell'intento di far giungere tutte le palline al nostro amico il più rapidamente possibile, lasceremo scivolare tutti i bits a gruppi di 7, cioé *carattere per*

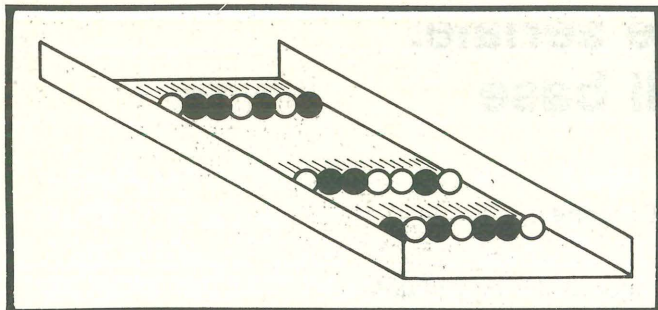



Figura 1 - Trasmissione in parallelo.

carattere. Abbiamo realizzato quella che è chiamata comunicazione *in parallelo* v. Figura 1. È necessario in questo caso, ricordiamolo, che il nostro cavo sia sufficientemente "largo". Se non possiamo disporre di un cavo siffatto, se cioé il nostro piano inclinato risulta tanto stretto da poter lasciar scorrere solo una pallina per volta, dobbiamo sfilare una pallina per volta da ciascun bastoncino e lasciarla cadere fino al nostro amico che a sua volta dovrebbe raccoglierla ed infilarla di nuovo nei suoi bastoncini a gruppi di 7. Infatti sia io che il mio amico non riusciamo a sopportare la presenza di palline sparse, ma solo infilate in bastoncini. Sia io che genero messaggi cioé (il microcalcolatore), sia il mio amico che li riceve (il terminale), riusciamo a lavorare solo con interi per volta (v. figura 2).
In conclusione possiamo dire che:

Si parla di trasmissione *in parallelo* quando è possibile trasmettere un carattere intero per volta.
Essa risulta più veloce ma richiede un cavo con tanti fili quanti sono i bits che costituiscono il carattere.
Si parla di trasmissione *seriale* quando la trasmissione di un carattere è effettuata un bit per volta in tempi successivi.
Essa risulta lenta e richiede un'operazione aggiuntiva di conversione parallelo-seriale e seriale-parallelo (in genere svolta dai dispositivi cosiddetti USART e/o UART).
Il vantaggio della trasmissione seriale è la possibilità di trasmettere economicamente su un unico filo.

D'ora in poi faremo sempre riferimento a terminali e micro entrambi interfacciati in modo *seriale*.
Un terminale che comunica in parallelo non è compatibile con uno che comunichi in seriale (a meno di usare gli USART menzionati)

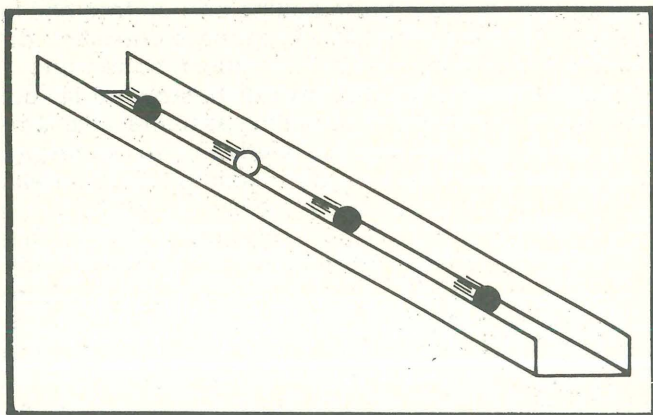



Figura 2 - Trasmissione seriale.

— baud - rate (velocità di trasferimento)

Continuando l'esempio precedente, una volta stabilito di avere a disposizione un piano inclinato tanto stretto da lasciar scorrere solo una pallina per volta e di essere costretti quindi ad una trasmissione seriale, posso pormi il problema di fissare una cadenza regolare di rilascio della pallina sul piano inclinato. Questo perché, una volta caduta la pallina il mio amico possa regolare la sua velocità di raccolta all'altra estremità del piano. Si capisce infatti che non dovrà, ad esempio, essere più lento di me, se non vuole che alcune palline gli cadano a terra.
Posso quantizzare la cadenza di rilascio delle palline contando il numero delle palline stesse che pongo sul piano in un intervallo di tempo convenzionale come il secondo (1 pallina al secondo = 1 bit al secondo = 1 baud). Se pongo sul piano 110 palline ogni secondo posso dire di trasmettere a 110 baud.
In genere gli attuali terminali seriali hanno baud-rate (velocità in baud) uguale ad uno dei seguenti valori:

110 baud
300 baud
600 baud
1200 baud
2400 baud
4800 baud
7200 baud
9600 baud

Alcuni terminali e calcolatori hanno la possibilità di collegarsi con un baud-rate a scelta fra tanti, selezionato opportunamente attraverso il posizionamento di ponticelli (dip-switches) e di commutatori rotativı (digit-switches).
Voglio evidenziare che, affinché un collegamento possa avvenire correttamente, dopo aver accertato che sia il trasmettitore che il ricevitore comunicano in seriale, è necessario che ambedue siano predisposti per lo stesso baud-rate.
Riassumendo:

Il *baud* è l'unità di misura della velocità di trasferimento dei dati su linea. Esso è uguale ad un bit/secondo.

Per chi volesse approfondire il discorso c'é da dire che questo non è sempre vero.

- sincrono / asincrono

Dopo aver definita la comunicazione come seriale e aver fissato una cadenza di trasmissione delle palline per ogni carattere (baud-rate), posso trovarmi a dover scegliere tra la possibilità di lasciare o non lasciare trascorrere un periodo di tempo tra la trasmissione di un gruppo di 7 palline (un carattere completo) ed il gruppo successivo. È possibile l'uno o l'altro caso a seconda che io disponga già di tutti i caratteri da trasmettere, tutti raccolti in una scatola tanto per intenderci, così che possa inviarli consecutivamente, oppure che sia costretto a cercare i bastoncini qua e là nella mia stanza (potrei essere una persona disordinata) e quindi a trasmetterli via via che li trovo. Nel primo caso sia io che il mio amico, come già detto,

dobbiamo poter disporre di una scatola di raccolta dei caratteri (buffer di input/output), nel secondo il buffer non è necessario. La prima, ancora, è una trasmissione che si dice di tipo *sincrono*, la seconda di tipo *asincrono*.
Il problema da risolvere è ora quello di fissare una convenzione per la sincronizzazione dei movimenti tra me e il mio amico. È possibile infatti che questi, ignorando l'istante in cui comincerò a trasmettere il messaggio, sia colto alla sprovvista dall'arrivo delle prime palline.
Se trasmetto in sincrono, posso risolvere il problema aggiungendo nella scatola che contiene i caratteri da trasmettere, prima o dopo, dei caratteri speciali (ad esempio palline *tutte* bianche oppure *tutte* nere). Questi caratteri speciali sono chiamati *caratteri di sincronismo*. Prima di cominciare a trasmettere i caratteri voluti, comincio quindi a trasmettere questi caratteri di sincronismo, sulla cui configurazione mi sono precedentemente accordato con il mio amico. Se egli è distratto, potrebbe non raccogliere le prime palline, ma dopo aver dato un'occhiata al piano ed aver visto arrivare queste palline, tutte bianche ad esempio, capirebbe che dopo i caratteri di sincronismo stanno per arrivare i primi caratteri validi: solo questi verranno acquisiti fino a quando non arriveranno i caratteri che segnalano la fine del messaggio. Questi ultimi saranno tralasciati.
Se io trasmetto in asincrono non è possibile ricorrere agli accorgimenti sopra riportati perché potendo variare il tempo che trascorre tra la trasmissione di un bastoncino ed il successivo, la trasmissione dei caratteri non avverrà ad un ritmo regolare e quindi c'è il rischio che il mio amico, tra un carattere e l'altro, possa distrarsi.
È necessario quindi fare in modo di destare la sua attenzione ogni volta che gli trasmetto un carattere, per esempio con l'invio, subito prima e subito dopo, di palline che non hanno altro significato che quello appunto di richiamare la sua attenzione (bits di *start* e di *stop*).

Riassumendo:

Mentre il baud-rate fa riferimento al tempo che intercorre tra la trasmissione di un bit ed il successivo nell'ambito di un carattere, i concetti di sincrono ed asincrono fanno riferimento al tempo che intercorre tra l'invio di un carattere ed il successivo.
Si parla infatti di trasmissione sincrona quando i caratteri che costituiscono il messaggio sono trasmessi in tempi immediatamente successivi. La sincronizzazione è in questo caso realizzata con l'invio, prima e dopo il messaggio, di caratteri speciali di inizio blocco, di sincronismo e di fine blocco.
Si parla invece di trasmissione asincrona quando non risulta definito il tempo che trascorre tra l'invio di un carattere e quello del successivo. La sincronizzazione è realizzata in questo caso con l'invio, prima e dopo il carattere, di bits di *start* e di *stop*.

La trasmissione asincrona così è meno veloce della sincrona per la necessità di rendere più lungo di almeno due bits ciascun carattere del messaggio.
Per quanto riguarda il numero dei bits di sincronizzazione in una comunicazione asincrona c'é da dire che, mentre il numero dei bits di start per ciascun carattere è fissato ad uno, risulta invece non fissato, dipendendo dal tipo di codice usato, il numero dei bits di stop (da uno a due). Anche per la caratteristica sincrono/asincrono è valida l'osservazione che sono interfacciabili tra di loro solo terminali e calcolatori entrambi sincroni od asincroni, e, nell'ambito dell'asincrono, in linea di massima, quelli che trattano caratteri con lo stesso numero di bits di stop.
È comunque possibile avere uno scambio corretto dei dati tra terminali che trasmettano caratteri con numero di bits di stop maggiore di uno e terminali predisposti a riceverne uno.
Alcuni terminali possono infine essere predisposti per trasmissioni asincrone di caratteri con numero di bits di stop variabile a seconda del posizionamento di ponticelli o interruttori.
In seguito faremo sempre riferimento a comunicazioni in asincrono.

— parity chek (controllo di parità)

Ci si riferisce qui ad un altro tipo di problema che s'incontra nella trasmissione digitale di dati: il controllo della validità della trasmissione. Ritornando all'esempio che ormai ha deciso di accompagnare ogni mia pretesa di spiegarvi qualcosa, dobbiamo dire che non basta che io ed il mio amico ci siamo accordati sul fatto di avviare una comunicazione seriale, a 110 baud, asincrona e con ciascun carattere preceduto da un bit di start e seguito da uno o due bits di stop. Può accadere infatti che le palline, rotolando lungo il piano, si scrostino, così che il mio amico, a volte, potrebbe trovarsi davvero imbarazzato nell'attribuire ad alcune palline il loro colore originale. Potrebbe esserci cioé un errore di trasmissione. Questo, si può capire, può avvenire tanto più facilmente quanto più il piano ha la superficie poco regolare, quanto più veloci scorrono le palline e quanto più lungo è il piano stesso. Gli errori di trasmissione cioè hanno una stretta dipendenza dalla bontà del cavo di connessione, dal baud-rate e dalla lunghezza del collegamento.
È possibile aiutare il mio amico ad accorgersi se c'é

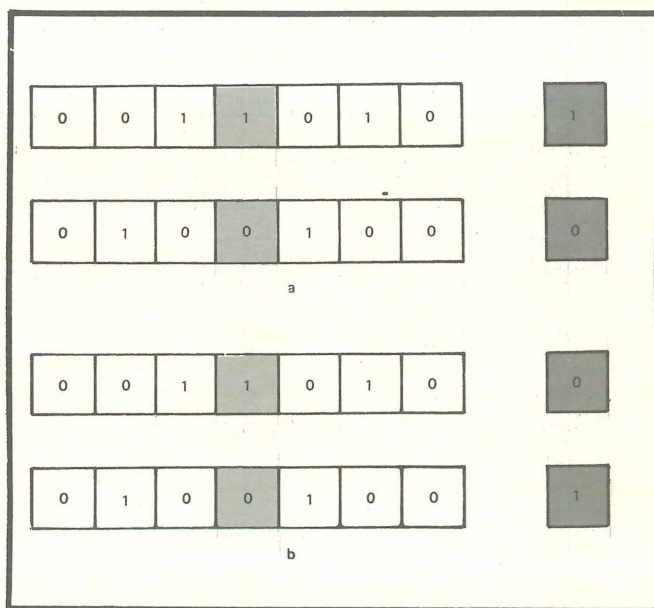



Figura 3 - Esempi di bit di parità a) "pari" e b) "dispari".

EBCDIC	Bit Configurazione	EBCDIC	Bit Configurazione	EBCDIC	Bit Configurazione	EBCDIC	Bit Configurazione
.	0100 1001	a	1000 0001	A	1100 0001	0	1111 0000
¢[	0100 1010	b	1000 0010	B	1100 0010	1	1111 0001
.	0100 1011	c	1000 0011	C	1100 0011	2	1111 0010
.	0100 1100	d	1000 0100	D	1100 0100	3	1111 0011
(	0100 1101	e	1000 0101	E	1100 0101	4	1111 0100
+	0100 1110	f	1000 0110	F	1100 0110	5	1111 0101
|	0100 1111	g	1000 0111	G	1100 0111	6	1111 0110
&	0101 0000	h	1000 1000	H	1100 1000	7	1111 0111
!]	0101 1010	i	1000 1001	I	1100 1001	8	1111 1000
$	0101 1011	j	1001 0001	J	1101 0001	9	1111 1001
*	0101 1100	k	1001 0010	K	1101 0010		
)	0101 1101	l	1001 0011	L	1101 0011		
	0101 1110	m	1001 0100	M	1101 0100		
¬	0101 1111	n	1001 0101	N	1101 0101		
-	0110 0000	o	1001 0110	O	1101 0110		
/	0110 0001	p	1001 0111	P	1101 0111		
.	0110 1011	q	1001 1000	Q	1101 1000		
%	0110 1100	r	1001 1001	R	1101 1001		
—	0110 1101	s	1010 0010	S	1110 0010		
	0110 1110	t	1010 0011	T	1110 0011		
?	0110 1111	u	1010 0100	U	1110 0100		
:	0111 1010	v	1010 0101	V	1110 0101		
#	0111 1011	w	1010 0110	W	1110 0110		
@	0111 1100	x	1010 0111	X	1110 0111		
.	0111 1100	y	1010 1000	Y	1110 1000		
	0111 1110	z	1010 1001	Z	1110 1001		
"	0111 1111						

Figura 4 - Codice EBCDIC. (Extended Binary Coded Decimal Interchange Code).

stato un errore di trasmissione (ma *non* a correggerlo) se, prima della trasmissione stessa, ci siamo ancora accordati sul fatto che, per ogni carattere da trasmettere, prima di ogni bit di stop, gli invio una pallina di un colore tale che il numero delle palline nere in suo possesso, per ogni carattere, risulti pari. Provvedo cioé alla trasmissione del cosiddetto *bit di parità* pari. (*V.R.C.* = Vertical *R*edundancy *C*heck = controllo di ridondanza verticale). Posso, al contrario, convenire con lui per l'invio di una pallina tale che il numero delle palline nere risulti dispari: in questo caso avrei concordato con lui un controllo di parità dispari (v. Figura 3).

Riassumendo:

> Per controllare la validità di trasmissione di un carattere in trasmissione asincrona si fa ricorso al bit di parità (V.R.C.).
> In genere si fissa una parità pari cioé il numero di bits "uno" del carattere (esclusi i bits di stop) risulta pari.

b7 b6 b5 — Bits					0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1
b4	b3	b2	b1	COLONNA / RIGA	0	1	2	3	4	5	6	7
0	0	0	0	0	NUL	DLE	SP	0	@	P	'	p
0	0	0	1	1	SOH	DC1	!	1	A	Q	a	q
0	0	1	0	2	STX	DC2	"	2	B	R	b	:
0	0	1	1	3	ETX	DC3	#	3	C	S	c	s
0	1	0	0	4	EOT	DC4	$	4	D	T	d	t
0	1	0	1	5	ENQ	NAK	%	5	E	U	e	u
0	1	1	0	6	ACK	SYN	&	6	F	V	f	v
0	1	1	1	7	BEL	ETB	'	7	G	W	g	w
1	0	0	0	8	BS	CAN	(	8	H	X	h	x
1	0	0	1	9	HT	EM	)	9	I	Y	i	y
1	0	1	0	10	LF	SUB	*	:	J	Z	j	z
1	0	1	1	11	VT	ESC	+	;	K	[	k	¦
1	1	0	0	12	FF	FS	,	.	L	\	l	|
1	1	0	1	13	CR	GS	—	-	M	]	m	¦
1	1	1	0	14	SO	RS	.	.	N	ˆ	n	
1	1	1	1	15	SI	US	/	?	O	—	o	DEL

Figura 5 - Codice ASCII (American Standard Code for Information Interchange).

Nel collegamento tra terminali dobbiamo verificare che ambedue effettuino un controllo di parità dello stesso tipo (entrambi parità pari o entrambi parità dispari). A questo proposito può essere utile l'informazione che per numerosi terminali esiste la possibilità di scegliere, sempre posizionando certi commutatori, tra un tipo di parità e l'altro.

— il codice

Questa è ancora un'altra caratteristica che deve essere accertata nel collegamento tra terminali. È abbastanza usuale che i messaggi scambiati tra terminali e calcolatori abbiano contenuto alfanumerico.È questo il caso, ad esempio, del terminale attraverso il quale dialoghiamo con il microcomputer. I comandi diretti al calcolatore e le sue risposte sono infatti parole e numeri.

Potendosi trasmettere in linea solo bits, evidentemente sia il terminale che il calcolatore devono possedere una tabella di conversione tra caratteri alfanumerici e la relativa configurazione in bits.

Benché le tabelle di conversione, cioé i *codici*, possono essere numerosi, convenzioni internazionali e l'uso corrente hanno portato a rendere più diffusi solo alcuni.

Si parla di codici a 5 bits, 6 bits, 7 bits, 8 bits, di codici cioé, ripetiamolo, che ad ogni carattere alfanumerico associano un carattere costituito da 5, o 6, o 7, o 8 bits. I più diffusi sono riportati nelle figure 4 e 5.

Il codice più usato, soprattutto nei terminali più recenti, risulta essere l'ASCII.

Anche per quanto riguarda i codici, dobbiamo accertarci che terminali e microcomputer comunichino nello stesso codice (tutt'e due ASCII, per esempio, o tutt'e due Baudot).

— simplex / half-duplex / full-duplex

Questi terminali fanno riferimento principalmente alla direzione del flusso dei dati su una linea di comunicazione.

Continuando l'esempio precedente delle palline verniciate, risulta evidente che le palline possono muoversi solo da me verso il mio amico se io abito in un appartamento più alto rispetto al suo. Si attua in questo caso una comunicazione di tipo *simplex*.

Se l'appartamento del mio amico fosse invece situato alla stessa altezza del mio, sarebbe al contrario possibile sia da parte mia fargli arrivare le palline con una spinta, sia da parte sua farmi arrivare le sue. È evidente che, essendo uno solo il cavo di collegamento, i dati si muoverebbero o in una direzione o nell'altra. Avremmo realizzato in questo caso una comunicazione di tipo half-duplex.

Se infine i cavi di collegamento fossero due, sarebbe possibile tra me ed il mio amico avviare una comunicazione simultanea nelle due direzioni. Si parlerebbe in questo caso di comunicazione in *full-duplex* (v. Figura 6).

Aggiungo, comunque, che non è strettamente necessario, pur realizzando una connessione in full-duplex, disporre di cavi distinti per le due direzioni del flusso di dati. Tecniche di trasmissione in "modulazione di portante" di solito fanno uso, per comunicazioni in full duplex, di un unico conduttore.

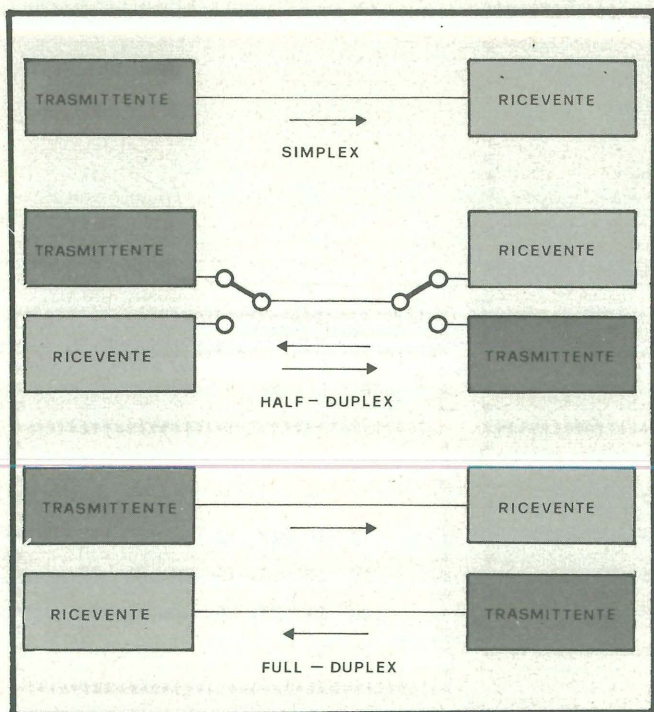



Figura 6 - Schemi di collegamento in simplex, half-duplex e full-duplex.

— banda base / modulazione di portante

È stato fatto riferimento in più punti al fatto che la comunicazione di dati a distanza avviene attraverso conduttori elettrici stesi tra il "trasmittente" ed il "ricevente". In un circuito chiuso siffatto i bits di informazione sono trasmessi variando una o più grandezze elettriche che definiscono il circuito stesso.
In particolare se a ciascun bit sono associati dei valori di grandezze di un circuito in corrente continua parliamo di trasmissione in *banda base*.
Di *modulazione di portante* invece in caso di circuito in corrente alternata.
In trasmissioni in banda base ai due valori 0 ed 1 sono associati due valori distinti della *tensione* oppure della *corrente*, in trasmissioni con modulazione di portante, in genere, valori distinti di *tensione*, *frequenza* o *fase*.
I concetti di banda base/modulazione di portante sono stati introdotti per spiegare meglio quanto accennato nel paragrafo precedente più che per la certezza che possano servirvi nella pratica: le connessioni tra terminali e microelaboratori sono in genere abbastanza brevi, per cui non si realizza in questi casi una comunicazione in modulazione di portante. Terminali e calcolatori inoltre si affacciano all'esterno sempre predisposti per collegamenti in banda base. Trasmissioni con modulazione di portante sono realizzate inserendo dispositivi esterni: i *modem*.

Con questo spero che i concetti fondamentali di comunicazione dati siano chiari:
Nella seconda parte parlerò degli standards più diffusi per comunicazione in banda base, in particolare Loop di corrente ed EIA RS 232 C. Illustrerò realizzazioni pratiche di interfacciamento generale di vario tipo e di interfacciamento particolare con riferimento a terminali e microcalcolatori tra i più usati. ■

```
CALENDARIO
DI BIT
DAL 1900 AL 1999
```

```
** 0 **** GENNAIO 1980 **** 366 **
 D  L  M  M  G  V  S
       1  2  3  4  5
 6  7  8  9 10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28 29 30 31
```

```
** 31 **** FEBBRAIO 1980 **** 335 **
 D  L  M  M  G  V  S
                1  2
 3  4  5  6  7  8  9
10 11 12 13 14 15 16
17 18 19 20 21 22 23
24 25 26 27 28 29
```

```
** 60 **** MARZO 1980 **** 306 **
 D  L  M  M  G  V  S
                   1
 2  3  4  5  6  7  8
 9 10 11 12 13 14 15
16 17 18 19 20 21 22
23 24 25 26 27 28 29
30 31
```

```
** 91 **** APRILE 1980 **** 275 **
 D  L  M  M  G  V  S
       1  2  3  4  5
 6  7  8  9 10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28 29 30
```

```
** 121 **** MAGGIO 1980 **** 245 **
 D  L  M  M  G  V  S
             1  2  3
 4  5  6  7  8  9 10
11 12 13 14 15 16 17
18 19 20 21 22 23 24
25 26 27 28 29 30 31
```

```
** 152 **** GIUGNO 1980 **** 214 **
 D  L  M  M  G  V  S
 1  2  3  4  5  6  7
 8  9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
29 30
```

```
** 182 **** LUGLIO 1980 **** 184 **
 D  L  M  M  G  V  S
       1  2  3  4  5
 6  7  8  9 10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28 29 30 31
```

```
** 213 **** AGOSTO 1980 **** 153 **
 D  L  M  M  G  V  S
                1  2
 3  4  5  6  7  8  9
10 11 12 13 14 15 16
17 18 19 20 21 22 23
24 25 26 27 28 29 30
31
```

```
** 244 **** SETTEMBRE 1980 **** 122 **
 D  L  M  M  G  V  S
    1  2  3  4  5  6
 7  8  9 10 11 12 13
14 15 16 17 18 19 20
21 22 23 24 25 26 27
28 29 30
```

```
** 274 **** OTTOBRE 1980 **** 92 **
 D  L  M  M  G  V  S
          1  2  3  4
 5  6  7  8  9 10 11
12 13 14 15 16 17 18
19 20 21 22 23 24 25
26 27 28 29 30 31
```

```
** 305 **** NOVEMBRE 1980 **** 61 **
 D  L  M  M  G  V  S
                   1
 2  3  4  5  6  7  8
 9 10 11 12 13 14 15
16 17 18 19 20 21 22
23 24 25 26 27 28 29
30
```

```
** 335 **** DICEMBRE 1980 **** 31 **
 D  L  M  M  G  V  S
    1  2  3  4  5  6
 7  8  9 10 11 12 13
14 15 16 17 18 19 20
21 22 23 24 25 26 27
28 29 30 31
```

Il calendario di BIT

di **Gloriano Rossi** (I2KH)

Un calendario pubblicato in primavera?? Che idea strana è poi questa!! Di strano in realtà ci trovo ben poco. Un calendario normalmente si propone verso la fine dell'anno!
È vero ma a qualche lettore, ne sono sicuro, interessa sapere come ad esempio si possa programmare l'individuazione di un anno bisestile oppure come determinare il giorno della settimana conoscendo l'anno, il mese ed il giorno.
Ma bando alle ragioni per le quali ...., e veniamo alla descrizione del nostro calendario.

Il programma

I primi passi del programma creano un bel riquadro,di presentazione sullo schermo video 'o sulla stampante collegata al sistema. In *riga 14* si potrebbe sostituire, anche se a noi della redazione non farebbe eccessivo piacere, la parola BIT con il vostro nome oppure potrebbe essere specificato, in caso di tabulato, il nome del futuro utilizzatore del calendario che in seguito verrà stampato.
Qualche pignolino potrebbe dire che la limitazione dall'anno 1900 all'anno 1999 risulta incongruente con l'affermazione di perpetuo, ma a riguardo posso smentire questo tipo di asserzione che a prima vista potrebbe sembrare fondata. Infatti le *righe 45* e *50* ci permettono di digitare esclusivamente le ultime due cifre dell'anno desiderato; quindi nessuno vieta di escluderle, obbligando d'altro canto a battere per intiero l'anno in questione, anche se questo fosse il 2035 od altro.
Lasciando inalterato il programma, qualsiasi numero compreso fra 00 e 99 viene automaticamente considerato come anno facente parte del secolo ventesimo, che per chi non lo sapesse è quello che và appunto dal 1900 al 1999. L'automatismo citato risiede esclusivamente nell'operazione di somma della cifra impostata più il numero 1900. Da questo punto in poi si può ben dire che il programma prosegue tutto da solo: stabilisce automaticamente se l'anno in questione è bisestile, individua il giorno della settimana del primo giorno del mese di gennaio e quindi stampa in conseguenza nella giusta posizione i vari giorni corrispondenti ai giorni del mese.
Stabilire se un anno è bisestile oppure no, potrebbe sembrare una cosa alquanto ardua e difficile, ma tutto ciò è falso; infatti qualsiasi anno non bisestile diviso per il numero 4 da come risultato un nuovo numero che sarà composto da una parte intera e sempre da una parte decimale. Come rilevare con un calcolatore la parte decimale? Semplice: si esegue la differenza tra anno diviso 4 e la parte intera, (senza cioè il decimale), della medesima operazione. Se il risultato è uguale a zero, quell'anno corrisponde ad un anno bisestile.
Vediamo, quale esempio, l'anno in corso: il 1980. Ebbene il numero 1980 diviso 4 da come quoziente 495,0 che sottratto alla sua stessa parte intera da come risultato zero.Invece con l'anno 1981 si ha:

(1981 : 4) — int (1981: 4) =
495,25 — int (495,25) =
495,25 — 495 = 0,25

Il risultato diverso da zero, ci dice che il 1981 non è bisestile.
Dopo aver eseguita questa operazione, il programma individua in che giorno della settimana cade il primo di gennaio. La routine richiamata con una semplice GOSUB 3000 svolge esattamente questo scopo.
L'obbiettivo di questa routine è quello di determinare un numero compreso fra 1 e 7 mediante una serie di operazioni le cui basi sono il giorno, il numero corrispondente al mese e l'anno espresso per intero.
Già a questo punto abbiamo le prime tre variabili che ci interessano: Y = 1980 che è bisestile (BIS = 1) e il primo di gennaio che è martedì in quanto è stato calcolato che D è uguale a 3 (1 = domenica, 2 = lunedì, 3 = martedì, ecc.).

Stampa dei mesi

In *riga 160* si inizia con un loop di lettura della tabella M che contiene i DATA di *riga 620*. L'elemento 2 della suddetta tabella M viene incrementato del valore contenuto nella variabile BIS che, come abbiamo già visto, per il 1980 è uguale ad 1 mentre per gli altri anni non bisestili sarà uguale a zero. Quindi riassumendo si può dire che dopo aver caricato i DATA in tabella M si incrementa il valore di M (2) (che normalmente risulta uguale al numero 28) con il valore contenuto in BIS che può essere zero oppure uno.
Inizia a questo punto l'inizializzazione e stampa del riquadro/mese dove troveremo, osservando da sinistra verso destra, un numero corrispondente al numero dei giorni trascorsi a partire dall'inizio dell'anno fino al primo giorno del mese che è in corso di stampa. Questo numero è calcolato incrementando il contatore S, e, in funzione della sua lunghezza di stringa, viene posizionato nel posto opportuno tramite le istruzioni di *riga 197, 198* e *199*. Dato che N risulta essere il numero del mese in esame (osservare le *righe 180* e *595*) tramite il comando BASIC: ON N GOTO, si andrà a prendere la stringa corrispondente al nome del mese da editare. A fianco di questo nome si stamperà Y che contiene, già dall'inizio e per tutta la durata del programma, l'anno del calendario che stiamo stampando. Si termina l'intestazione del mese calcolando e stampando quanti giorni mancano dal

Philips. Alimentatori da banco da 20 a 120W.

Gli alimentatori da banco Philips lavorano:

- a lungo: MTBF= 75.000 ore
- in condizioni di lavoro avverse: norme IEC-68 testi Fc e Eb
- in serie ed in parallelo
- inoltre sono regolabili in tensione ed in corrente
- hanno una eccellente stabilità ed una grande affidabilità

Se volete ulteriori informazioni richiedeteci la documentazione completa.

Tipo		PE1535	PE1536	PE1537	PE1538	PE1539	PE1540	PE1541	PE1542	
Uscite	V	0-40V	0-20V	0-40V	0-75V	0-20V	0-40V	0-75V	0-7V	2x0-20V
	A	0-0,5A	0-2A	0-1A	0-0,5A	0-6A	0-3A	0-1,6A	0-3A	0-1A
Stabilità		≦0,5%	≦0,01%	≦0,01%	≦0,01%	≦0,01%	≦0,01%	≦0,01%	≦0,05%	≦0,05%
Ripple		≦1,5mV	≦1mV	≦1mV	≦1mV	≦1mV	≦1mV	≦1mV	≦0,5mV	≦1mV
Coefficiente di temperatura		≦0,02%/°C	≦0,01%/°C	≦0,01%/°C	≦0,01%/°C	≦0,01%/°C	≦0,01%/°C	≦0,01%/°C	≦0,02%/°C	≦0,02%/°C
Tempo di ripristino		≦10 μs	≦25 μs	≦25 μs	≦25 μs	≦50 μs	≦50 μs	≦50 μs	≦50 μs	≦50 μs

Sede: Philips S.p.A. - Reparto T&M - V.le Elvezia, 2 - 20052 MONZA - Tel. (039) 3635249

Filiali: Torino tel. (011) 210404 - Padova (049) 657700 - Bologna (051) 712054 - Roma (06) 382041
Palermo (091) 400066 - Cagliari (070) 560761

Strumenti Elettronici di Misura

PHILIPS

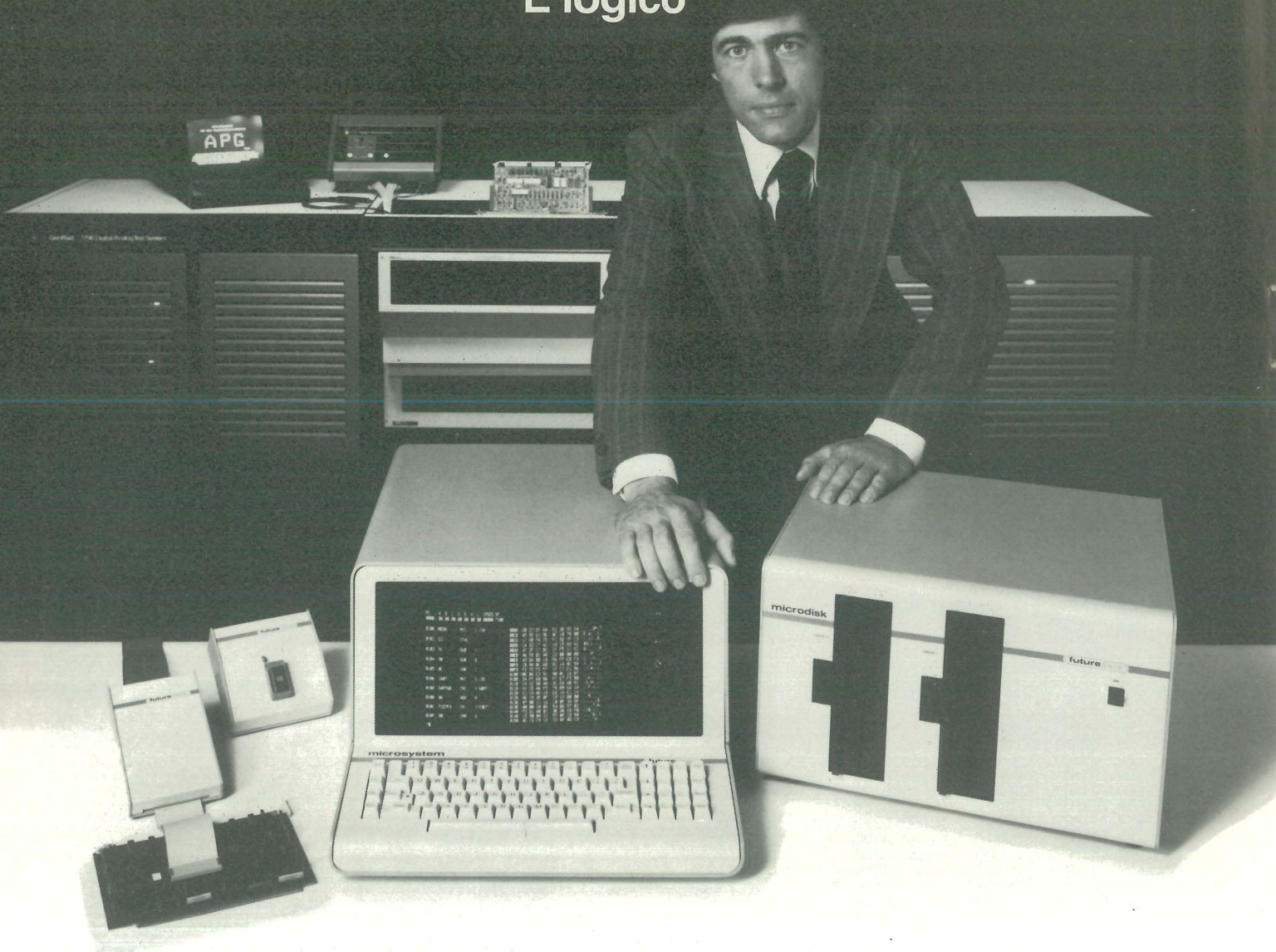
```
        CALENDARIO  DI  BIT

0 CLR:RESTORE
1 REM CALENDARIO PERPETUO DA 1900 A 1999
2 REM EDIZIONE ELABORATA PER LA RIVISTA :
3 REM
4 REM              B I T
5 REM
6 REM DA GLORIANO ROSSI (I2KH)
7 REM C.SO PORTA NUOVA 46 - 20121 MILANO
8 REM************************************
10 FORI=1TO39:PRINT"*";:NEXT:PRINT
11 FORI=1TO7:PRINT"*"TAB(38)"*":NEXT
12 PRINT"*        C A L E N D A R I O        *"
13 PRINT"*"TAB(38)"*"
14 PRINT"*             DI   B I T            *"
15 PRINT"*"TAB(38)"*"
16 PRINT"*         DAL 1900 AL 1999          *"
20 FORI=1TO7:PRINT"*"TAB(38)"*":NEXT
21 FORI=1TO39:PRINT"*";:NEXT:PRINT
30 PRINT:PRINT
40 PRINT"CHE ANNO VUOI  "::INPUTY
45 IFY>99ORY<0THEN40
50 Y=1900+Y
110 DIMM(12)
120 FORI=1TO6:PRINTCHR$(10);:NEXTI
130 GOSUB2000
135 GOSUB3000
140 S=0
150 P1=0:IFBIS=1THENP1=13
160 FORN=0TO12:READM(N):NEXTN
170 M(2)=M(2)+BIS
180 FORN=1TO12
190 PRINT:PRINT:S=S+M(N-1)
195 GOSUB1000
196 FORI=1TO39:PRINT"*";:NEXT:PRINT
197 SP$="":IFS<10THENSP$="  "
199 IFS<100ANDS>9THENSP$=" "
200 PRINT"**"SP$;S;TAB(7);
210 FORI=1TO4:PRINT"*";:NEXTI
220 ONNGOTO230,240,250,260,270,280,290,300,310,320,330,340
230 PRINT"  GENNAIO  ";:GOTO350
240 PRINT"  FEBBRAIO ";:GOTO350
250 PRINT"  MARZO    ";:GOTO350
260 PRINT"  APRILE   ";:GOTO350
270 PRINT"  MAGGIO   ";:GOTO350
280 PRINT"  GIUGNO   ";:GOTO350
290 PRINT"  LUGLIO   ";:GOTO350
300 PRINT"  AGOSTO   ";:GOTO350
310 PRINT" SETTEMBRE ";:GOTO350
320 PRINT"  OTTOBRE  ";:GOTO350
330 PRINT" NOVEMBRE  ";:GOTO350
340 PRINT" DICEMBRE  ";
350 PRINTY;
355 FORI=1TO4:PRINT"*";:NEXTI
360 S2$="":IF365-S<100THENS2$=" "
365 NG=365+BIS
370 PRINTS2$;NG-S;"**"
375 FORI=1TO39:PRINT"*";:NEXT
376 PRINT:PRINT"*"SPC(37)"*"
380 PRINT"*      D   L   M   M   G   V   S      *"
400 PRINT"*"SPC(37)"*"
410 FORI=1TO39:PRINT"*";:NEXTI
420 PRINT
430 FORW=1TO6
440 PRINT"*"SPC(37)"*"
450 PRINT"*    ";
460 REM
470 FORG=1TO7
480 D=D+1
490 D2=D-S
500 IFD2>M(N)THENPRINTTAB(38)"*":GOTO580
510 IFD2>0THENPRINTD2;
515 P=0:IFD2<10THENP=1
520 PRINTTAB(P+4+4*G);
530 NEXTG
535 PRINT"     *"
540 REM
550 IFD2=M(N)THEN590
560 NEXTW
570 REM
580 D=D-G
590 PRINT"*"SPC(37)"*"
591 FORI=1TO39:PRINT"*";:NEXTI:PRINT
595 NEXTN
600 REM
610 FORI=1TO6:PRINTCHR$(10);:NEXTI
620 DATA0,31,28,31,30,31,30,31,31,30,31,30,31
640 END
1000 IFN=1THEN1020
1005 PRINT"BATTERE -1- PER PROSSIMO MESE,-0- FINE"
1010 INPUTX$:IFX$<>"1"THENEND
1020 PRINTCHR$(147):RETURN
2000 BIS=0
2010 X=Y/4:X1=INT(Y/4)
2020 IFX-X1=0THENBIS=1
2030 RETURN
3000 D=1:M=1
3010 K=INT(0.6+(1/M))
3020 L=Y-K
3030 O=M+12*K
3040 P=L/100
3050 Z1=INT(P/4)
3060 Z2=INT(P)
3070 Z3=INT((5*L)/4)
3080 Z4=INT(13*(O+1)/5)
3090 Z=Z4+Z3+(-Z2)+Z1+D-1
3100 D=Z-(7*INT(Z/7))+1
3110 D=(D-1)*-1
3120 RETURN
```

primo giorno del mese in corso fino alla fine dell'anno (*riga 360, 365* e *370*).
Subito dopo viene eseguita una semplice stampa di righe con il contenuto fisso delle abbreviazioni dei giorni della settimana, e quindi si ha la routine (costituita da un loop ripetuto sei volte) che stampa i numeri corrispondenti ai giorni del mese nelle opportune colonne dei giorni della settimana. La routine in oggetto comprende le *righe 430* e *560*. Abbastanza interessanti sono le istruzioni di riconoscimento di fine del mese e quelle di giusta tabulazione (*righe 500, 510* e *520).*
Il listato proposto è abbastanza semplice e contiene istruzioni in linguaggio BASIC standard, per cui non dovrebbero esserci grossi problemi per chi ha il BASIC non esteso.
Qualche istruzione potrebbe essere cambiata per l'uso diretto su stampante: istruzioni tipo TAB o PRINT CHR$(10) potranno essere sostituite con SPC e con PRINT.

Questo programma si presta facilmente ad aggiunte e modifiche. Infatti è possibile, aggiungendo qualche istruzione, ottenere direttamente un solo mese desiderato, senza dover essere obbligati a far passare le stampe dei singoli mesi precedenti. Un obbiettivo di questo genere potrebbe essere risolto individuando innanzitutto il giorno della settimana in cui cade il primo giorno del mese richiesto, e quindi entrando nel loop di stampa / mese ed uscirne con un test opportuno prima del comando NEXT N.
Un'altra aggiunta potrebbe essere quella di implementare domande quali ad esempio:

— Vuoi un calendario? RPL = 0 —
— Che giorno era? RPL = 1 —
— Bisesto o no? RPL = 2 —

e quindi agire di conseguenza utilizzando le routines che sono contenute nel programma.
Insomma c'è da divertirsi e ..... dando un pò di numeri ottenere un programmino di sicuro effetto.
3500 CLR: RESTORE:END

È logico

Il sistema di sviluppo a microcalcolatore più avanzato nel mondo viene dal leader mondiale dei sistemi di collaudo

La GenRad, più di qualsiasi altro, ha una lunga esperienza nel fornire soluzioni complete per problemi di collaudo elettronici.
Ora con il sistema di sviluppo universale a microcalcolatore GenRad/Futuredata 2300, noi utilizziamo questa esperienza e queste capacità per aiutare il progettista di sistemi a risolvere le sue esigenze e ad accelerare il completamento del suo progetto.
Il sistema GenRad/Futuredata 2300 riduce il tempo d'inserzione del programma usando una scrittura del testo sul video interattivo, con una struttura di comandi facili da usare.
Il sistema permette anche uno sviluppo software strutturato, con veloce microassembler riallocabile, linguaggio di supporto ad alto livello in Basic e Pascal, potenti caratteristiche di collegamento ed individuazione.
Il progettista può verificare l'esecuzione del software con debugger (revisione di programma per individuazione errore) a schermo suddiviso. Il disassemblaggio del programma, il debugging simbolico, le caratteristiche di analisi logica sono tutte parti del sistema.
L'integrazione del software e dell'hardware è realizzata con simulazione in tempo reale in-circuit che supporta l'intera memoria e rappresenta gli ingressi/uscite.
La struttura di approccio universale del GenRad/Futuredata 2300 copre una ampia gamma di processori comprendente il supporto hardware e software per l'8086 e lo Z8000.
Assicura la compatibilità completa alle progettazioni future.
Da ultimo, non comunque in ordine di importanza, il GenRad/Futuredata 2300 si accorda con i vostri fabbisogni di espansione.
La sua configurazione rete supporta fino a otto utilizzatori simultaneamente, suddividendo dischi e risorse periferiche.
Fate che il successo sia parte del vostro prossimo progetto.
Chiedete maggiori informazioni sui prodotti GenRad/Futuredata telefonando o scrivendo a:

DIVISIONE DELLA

Per maggiori informazioni vi invitiamo a contattare la GenRad S.p.A. Via Lampedusa 13 - 20141 Milano - Tel. 02-8466541 - Telex 320373 - Ufficio di Roma - Tel. 06-4384155.

Tester per circuiti integrati TTL realizzato con il Nanocomputer NBZ80-S

Parte II di **A. Cavalcoli**

Nel precedente articolo (Parte I, BIT N.6, pag. 99) si è parlato della configurazione del sistema didattico Nanocomputer NBZ80-S della SGS-ATES e della teoria dell'applicazione. In questa seconda parte verrà invece descritta in dettaglio l'applicazione in oggetto.

Interfacciamento

I collegamenti per la realizzazione del nostro tester sono riportati in *Figura 1*.

Il PIO N.2 del Nanocomputer è interfacciato con il chip sotto test tramite gli zoccoli che rendono disponibili sulla breadboarding station i segnali del PIO stesso.

Due porte del PIO, Porta C e Porta D, sono collegate ai 16 pin di un DIP standard la cui posizione viene usata sia per il circuito di riferimento che per quello sotto test. Nel caso di un chip a 14 pin, i pin 8 e 9, non usati, sono posti a massa.

Facendo riferimento a quanto detto nel precedente articolo (pag 100 e 101) relativamente a "Il circuito di I/O parallelo PIO", si ricorda che le Porte C e D devono operare in modo 3 (modo di controllo dei singoli bit): le 16 linee dati delle porte infatti lavorano in parte come input, in parte come output.

Vale quanto segue:

— bit Dn eguale a 0 nel byte di controllo della Porta C indica che la linea PCn è di uscita
— bit Dn eguale a 1 nel byte di controllo della Porta C indica che la linea PCn è di ingresso
— bit Dn eguale a 0 nel byte di controllo della Porta D indica che la linea PDn è di uscita
— bit Dn eguale a 1 nel byte di controllo della Porta D indica che la linea PDn è di ingresso

Alcune precauzioni

La configurazione del byte di controllo deve essere specificata dall'utente; nel nostro caso occorre fare attenzione ad alcuni particolari:

— I pin di alimentazione e di massa del circuito sono da considerarsi come ingressi al Nanocomputer, quindi nel byte di controllo deve esserci un 1 in corrispondenza della linea della porta del PIO.

COLLEGAMENTI ZOCCOLO C (A) per il test di componenti a 16 pin

PIO	Collegamento
PC0	Pin 1
PC1	Pin 2
PC2	Pin 3
PC3	Pin 4
PC4	Pin 5
PC5	Pin 6
PC6	Pin 7
PC7	Pin 8
CRDY	
CSTB	
PD0	Pin 9
PD1	Pin 10
PD2	Pin 11
PD3	Pin 12
PD4	Pin 13
PD5	Pin 14
PD6	Pin 15
PD7	Pin 16
DRDY	
DSTB	

COLLEGAMENTI ZOCCOLO C (B) per il test di componenti a 14 pin

PIO	Collegamento
PC0	Pin 1
PC1	Pin 2
PC2	Pin 3
PC3	Pin 4
PC4	Pin 5
PC5	Pin 6
PC6	Pin 7
PC7	0
CRDY	
CSTB	
PD0	0
PD1	Pin 8
PD2	Pin 9
PD3	Pin 10
PD4	Pin 11
PD5	Pin 12
PD6	Pin 13
PD7	Pin 14
DRDY	
DSTB	

Figura 1 - Schema del circuito del tester per i circuiti integrati.

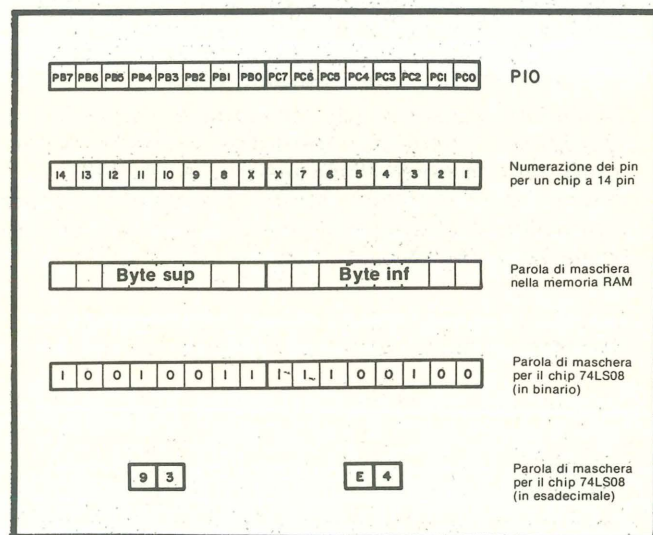



Figura 2 - Byte di controllo della configurazione di I/O per i quattro dispositivi AND 74LS08.

Circuiti integrati	Parola di maschera	
	Byte superiore	Byte inferiore
74LS02	C9	C9
74LS04	AB	EA
74LS05	AB	EA
74LS08	93	E4
74LS30	CF	C0
74LS32	93	E4
74LS42	87	FF
74LS125	93	E4
74LS139	8F	F8
74LS365	95	D4

Tabella I - Byte di controllo della configurazione di I/O per alcuni chips della serie 74LSXX.

— I pin di ingresso al chip sotto test sono delle uscite dal Nanocomputer per cui si avrà in corrispondenza uno 0 nel byte di controllo.
—, I pin di uscita del chip sotto test sono ingressi al Nanocomputer, per cui si avrà in corrispondenza un 1 nel byte di controllo.
— I pin NC, non collegati, del chip sotto test devono essere messi a massa, per cui vale quanto detto al primo punto.

Una volta individuata la configurazione dei due bytes di controllo (detti anche maschere) per le due porte del PIO N. 2, l'utente comunica questa informazione al tester tramite due locazioni di memoria, indicate nel nostro software (v. oltre) come MASKW e MASKW + 1: il contenuto di tali locazioni dev'essere inserito dall'utente tramite la tastiera dell'NBZ80-S (MASKW per la Porta C e MASKW + 1 per la Porta D). Riassumendo, ogni pin di uscita per il chip sotto test è un *ingresso* nel Nanocomputer, da cui l'assegnazione di un corrispondente 1 logico; viceversa ogni pin di ingresso del chip sotto test è una *uscita* dal Nanocomputer, da cui uno 0 logico. Con calma ed attenzione ci si abituerà ad utilizzare senza patemi d'animo questi potenti "LSI attorno al micro", di cui il PIO è un rappresentante.

I bytes di controllo

Le diverse configurazioni dei bit d'ingresso e di uscita del Nanocomputer formeranno una tabella della verità (come quella rappresentata nella *Figura 6* del precedente articolo), che è relativa al tipo di chip sotto il controllo, e che viene generata e gestita dalla routine CHPTST, che vedremo oltre.

Nella *Figura 2* sono indicate, come esempio, le configurazioni binarie dei due bytes di controllo per un chip del tipo porta AND quadrupla 74LS08.

Nella *Tabella I*, invece, sono rappresentati i bytes di controllo, in esadecimale, relativi ai più usuali chips della serie 74LSXX.

Il software necessario

Come già può essere intuito, l'operatività del tester è legata non tanto all'hardware quanto al software inserito nel Nanocomputer.

Nella *Tabella II* diamo, per vostra documentazione, il software originario, tratto dal testo *Nanobook N. 3*, edito dalla Jackson Italiana.

A questo punto occorre entrare nel dettaglio di questo software, per capire come vengono utilizzati lo Z80 e il Nanocomputer NBZ80-S.

```
Codice     Codice
Oggetto    Sorgente                                Commenti

3E03                  NAME CHPTST
           CHPTST:    LD A,03H          ;Reset del Flip-Flop di
D30A                                    ;abilitazione interrupt del PIO
D30B                  OUT (0AH),A
2A0300                OUT (0BH),A
                      LD HL, (MASKW)    ;Accetta la maschera per il circ.
010AFF
ED41                  LD BC,0FF0AH
ED69                  OUT (C),B         ;Porta A del PIO al modo 3
                      OUT (C),L         ; maschera I/O per la Porta A

OC
ED41                  INC C             ;Cambia alla Porta B
                      OUT (C),B         ;Porta B del PIO al modo 3
ED61
                      OUT (C),H         ;Maschera I/O per la porta B

31A00F     ;
           REF·       LD SP,CHPSTK      ;Inizializza il
DD210008                                ;puntatore dello stack
                      LD IX,REFIC       ;Inizializza il
                                        ;puntatore della
010000                                  ;mappa dell'IC di riferimento
                      LD BC,0000H       ;Inizializza
CD8806                                  ;la parola del contatore
                      CALL STORE        ;Genera la
00                                      ;tabella di riferimento
           ENDREF:    NOP
31A00F     ;
           UNKN:      LD SP,CHPSTK      ;Inizializza
DD21000C                                ;il puntatore dello stack
                      LD IX,UNKIC       ;Inizializza
                                        ;il puntatore
                                        della mappa dell'IC incognito
010000                LD BC,0000H       ;Inizializza
                                        ;la parola del contatore
```

Tabella II

Dettaglio del software

Con l'etichetta CHPTST si indica l'inizio del programma per il tester. In REFIC, fino REFIC + 03FFH, è allocata un'area per memorizzare i bit di uscita dalla tabella della verità generata utilizzando il circuito integrato di *riferimento*.

Da UNKIC a UNKIC + 03FFH è allocata l'area per i bit di uscita dalla tabella della verità generata utilizzando il circuito integrato sotto test (quello cioè sconosciuto).

CHPSTK è il valore iniziale dello stack pointer.

Vediamo meglio le cose facendo riferimento alla tabella II.

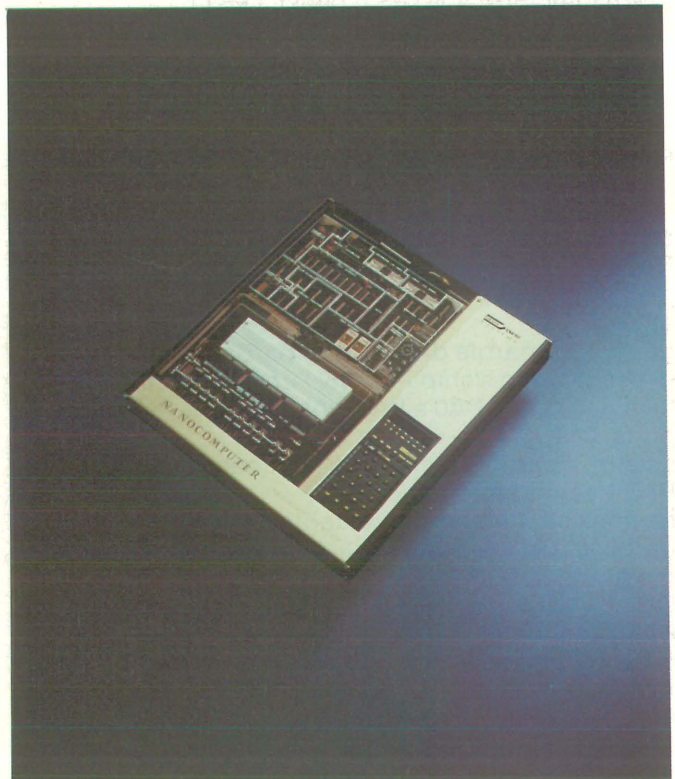
CHPTST parte con l'inizializzazione delle due porte C e D del PIO N. 2, secondo le modalità indicate precedentemente.

Le routines REF e UNKN sono le routines di inizializzazione dei registri rispettivamente per il chip di riferimento e per quello sconosciuto: queste routines inizializzano lo stack pointer e un registro puntatore verso l'area di memoria in cui deve essere memorizzata la tabella di risposta del circuito in esame.

```
CD8806            CALL STORE      ;Genera la
                                  ;tabella d'uscita
                                  ;degli IC incogniti
          ;
210008  COMPAR:   LD HL,REFIC     Preparati al confronto
                                  ;impiegando
                                  ;l'istruzione CPI
11000C            LD DE,UNKIC     ;HL punta alla
                                  ;tab. di rifer, DE
                                  ;punta alla tabella
                                  ;dell'IC incognito
1A      NEXTB:    LD A,(DE)       ;Carica nell'accumulatore
                                  ;il byte d'uscita
                                  ;incognito
EDA1              CPI             ;Confronta con (HL)
2037              JR NZ,BAD       Se non =, si ha
                                  ;un IC non buono
13                INC DE          Se =, preparati per il test
                                  ;del byte successivo
EA7D06            JP PE,NEXTB     ;Se il flag P/V=1
                                  ;vai al test del byte successivo
1833    GOOD:     JR START        ;Se il flag P/V = 0
                                  ;BC è zero
                                  ;e si è eseguito
                                  ;il test di tutti i bytes
                                  ;Inizializza la
          ;
110000  STORE:    LD DE,0000H     ;parola del test
2A0300  NTEST:    LD HL,(MASKW)   ;Carica HL con la
                                  ;parola della maschera
7B                LD A,E          ;Esegui l'AND a 16 bit
                                  ;sulla maschera e
                                  ;sulle parole di test
A5                AND L
6F                LD L,A
7A                LD A,D
A4                AND H
67                LD H,A
7C      MASK:     LD A,H          ;Controlla se il risultato
                                  ;dell'AND a 16 bit è = 0
B5                OR L
201B              JR NZ,NXTWD     ;Se non è 0 vai al
                                  ;byte di test successivo
          ;
7B      TEST:     LD A,E          ;Se =0, è una
                                  ;parola di test valida.
                                  ;Falla uscire sull'IC
D308              OUT (08H),A
7A                LD A,D
D309              OUT (09H),A
2A0300            LD HL,(MASKW)   ;Accetta la parola della
                                  ;maschera per IC
DB08              IN A,(08H)      Ingresso del byte LO
                                  ;dall'IC
A5                AND L           ;Mascheralo
DD7700            LD (IX),A       ;Memorizzalo
DD23              INC IX          ;Aggiorna IX
DB09              IN A,(09H)      ;Ingresso del byte HI
                                  ;dall'IC
A4                AND H           ;Mascheralo
DD7700            LD (IX),A       ;Memorizzalo
DD23              INC IX          ;Aggiorna IX
03                INC BC          ;Somma due al
                                  ;contatore
03                INC BC
          ;
13      NXTWD:    INC DE          ;Accetta la parola di test
                                  ;successiva
7A                LD A,D
B3                OR E
20D3              JR NZ,NTEST     ;Se DE non è zero
                                  ;ritorna indietro
                                  ;parola di test
C9                RET             ;Se DE è zero
                                  ;viene generata l'uscita
                                  ;dell'intera tabella
1800    BAD:      JR START        ;IC non buono,
                                  ;riparti
          ;
18AD    START:    JR UNKN         ;Salta alla routine
                                  ;di test per IC
                                  ;incognito
```

Viene poi inizializzata la coppia di registri BC, che sono utilizzati per il conteggio dei bytes memorizzati nella tabella di risposta.
La routine COMPAR realizza il confronto, byte per byte, delle due tabelle di risposta, quella del chip di riferimento e quella del chip sotto test.
Vengono poi posti due breakpoints alle locazioni di memoria GOOD e BAD, in modo da distinguere i due possibili risultati del test effettuato.
In totale sono usati i seguenti registri: BC, DE, HL, IX.
La subroutine STORE inizializza la parola di test a 0000, carica i bytes di controllo ed attua un'operazione AND, di 16 bit, tra la parola di test e quella di controllo.
La routine MASK determina se tutti i 16 bit di risultato dell'operazione di AND sono posti a 0: se questo è

Parola di test	Parola di maschera 74LS08	Test • Maschera	Parola di testo valida?
00000000 00000000	10010011 11100100	00000000 00000000	si
00000000 00000001	10010011 11100100	00000000 00000000	si
00000000 00000010	10010011 11100100	00000000 00000000	si
00000000 00000011	10010011 11100100	00000000 00000000	si
00000000 00000100	10010011 11100100	00000000 00000100	no
00000000 00000101	10010011 11100100	00000000 00000100	no
00000000 00000110	10010011 11100100	00000000 00000100	no
00000000 00000111	10010011 11100100	00000000 00000100	no
00000000 00001000	10010011 11100100	00000000 00000000	si

Tabella III - Esempio di operazione di AND tra le parole di test e di maschera.

Il Nanocomputer NBZ80-S della SGS-ATES.

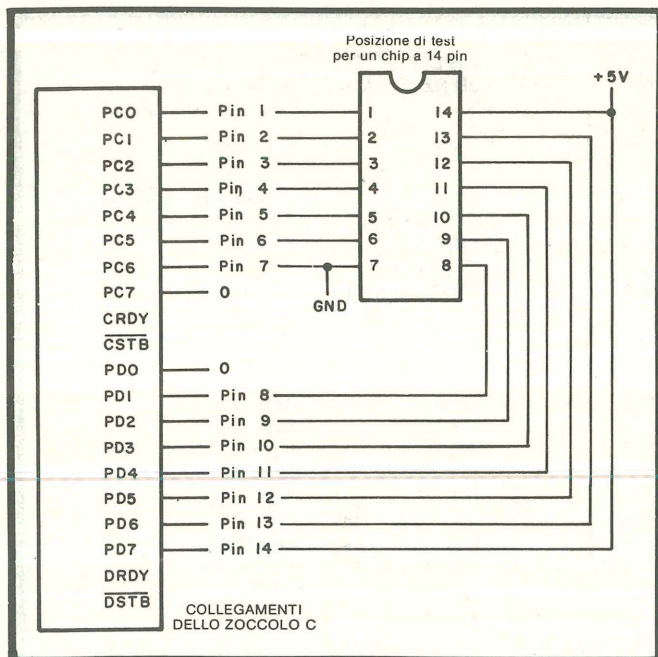



Figura 3 - Schema del circuito per l'esperimento presentato.

vero, la parola di test, presente nella coppia di registri DE, è una parola di test valida che può essere posta in uscita verso le porte C e D del PIO.
Dopo l'operazione di uscita, sono letti, uno dopo l'altro, due bytes, mascherati e poi allocati nella tabella delle risposte. Il puntatore, il registro IX, è incrementato due volte. Pure il contatore è incrementato due volte.
Alla fine, la parola di test, che deve andare da 0000 a FFFF, è ancora incrementata; si verifica se si è completato l'esame, cioè se si è tornati a 0000: se no, si ritorna nel loop a NTEST (Next TEST).

Esempio

Nel caso di un 74LS08, i bytes di controllo sono:
10010011
11100100
cioè in esadecimale, 93 E4.
Nella *Tabella III* è presentata una serie di parole di test, da 0000H a 0008H, ed il corrispondente risultato all'AND.
Le prime 4 parole di test sono parole valide da porre in uscita verso il chip 74LS08; l'unico cambiamento di bit è nelle linee PA0 e PA1, che sono due ingressi alla porta AND G1 (si veda la *Figura 2*).
Le altre 4 parole di test sono eliminate perchè la linea PA2 ha uno stato logico 1.
La linea ed il bit corrispondente PA2 non può essere un bit di uscita dal microcomputer, come del resto mostrano le tabelle finora presentate nell'articolo.
Quindi, tutte le parole di test che contengono un 1 in PA2 possono essere eliminate, dato che un 1 od uno 0 nel bit di uscita non influenzano il test.
Applicando la stessa regola a PA5, PA6, PA7, PB0, PB1, PB4, PB7, si selezionano solo 256 parole di test valide tra le possibili 65.536. Si noti che 256 sono le differenti combinazioni logiche degli 8 ingressi al chip 74LS08.

L'esperimento

Può essere interessante a questo punto seguire passo passo un ipotetico utente del Nanocomputer NBZ80-S nell'esecuzione dell'esperimento fin qui descritto.
Si voglia, allora, sperimentare l'efficacia del nostro tester utilizzando un chip 74LS08, porta AND a due ingressi quadrupla.
Si faccia riferimento alla *Figura 3*.
La sequenza è:

— Caricamento del programma CHPTST.
(Ricordiamo che tutti i programmi relativi ad esperimenti realizzabili con il Nanocomputer NBZ80-S e descritti nel Nanobook N. 3, come appunto il presente esperimento, possono essere acquistati dalla SGS-ATES già memorizzati su ROM).
Per caricarlo basta fare:
a) caricare nel Program Counter, da tastiera, l'indirizzo F000
b) premere il tasto di GO. Sul display appare in sequenza:

SGS-ATES NANO ROUTINES RELEASE 2-2
LOADED CIAO

I programmi, tra cui CHPTST, sono a questo punto caricati nella memoria RAM del Nanocomputer.

— Caricamento delle maschere.
a) caricare in MASKW, locazione 0003, il valore E4, byte di maschera "low" per il chip 74LS08
b) caricare in MASKW + 1, locazione 0004, il valore 93, byte di maschera "high" per il chip 74LS08

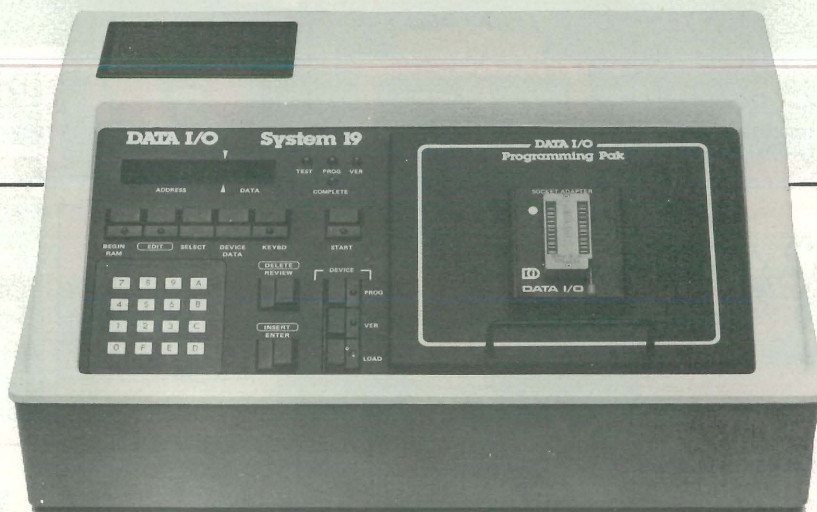
— Caricare tre punti di breakpoint alle seguenti locazioni:
0669: ENDREF per la fine di REF
0686: GOOD , per i test con risultato "OK"
06B9: BAD , per i test con risultato negativo

— Si inserisca il circuito integrato di riferimento, il 74LS08 nella posizione di test e si inizi l'esecuzione del programma dalla locazione CHFTST. AL termine dell'esecuzione, si è generata una tabella di riferimento per il chip 74LS08.

— Si tolga il 74LS08 "noto" dalla sua sede, e si collochi invece un altro chip, ad esempio un 74LS32, nella posizione di test.

— Si inizi l'esecuzione del programma a partire da ÙNKN. Si osserva che il progrmma si arresta in corrispondenza del breakpoint BAD, locazione 06B9, questo perchè la tabella generata non coincide con quella, già memorizzata, di riferimento.

A questo punto l'utente dell'NBZ80-S può verificare nella pratica le situazioni più diverse.
Si noti che da questo semplice esperimento è comunque risultato evidente il grosso peso del software e l'aspetto quasi inessenziale delle parti hardware. Questa realtà la incontrerete costantemente: una tendenza costante alla crescita delle componenti software nella progettazione di sistemi a microprocessore. ■

"BEEP! BEEP!"

Il programmatore universale di Prom DATA I/O risponde per farti risparmiare tempo e denaro.

È un programmatore che ti dice che tutto va bene o, se qualche cosa non va, ti specifica perchè.

Accendilo ed il Mod. 19 prova tutti i suoi componenti: porte input/output, microprocessore, RAM, Bus, pannello frontale e persino il suo software interno.

Durante la sequenza di programmazione il sistema 19 esegue il check dell'"illegal bit", il "blank check", il "sum check" del componente appena programmato e la verifica a tensione alta, bassa, media come richiesto per essere sicuri che il componente soddisfi le specifiche operative del costruttore.

Il sistema 19 avvisa in modo discreto l'operatore quando è stato fatto un semplice errore di procedura e si blocca prima che l'operatore possa fare un errore più serio e possa danneggiare il componente.

Se qualche cosa va male durante le operazioni, il sistema 19 lampeggia indicando sul display uno dei 27 codici di errore in modo da guidarti velocemente alla soluzione del problema specifico.

Con il sistema 19 potrai programmare ogni tipo di componente logico programmabile comprese le FPLA. Con il suo modulo multiplo potrai programmare fino ad otto componenti MOS alla volta.

La cosa migliore è che il DATA I/O Sistema 19 ha un prezzo abbordabile da tutti.

È impossibile fare una ricerca di mercato sui programmtori senza considerare il mod. 19.

Fa in modo che possiamo spiegarti le differenze, telefona o scrivi alla SISTREL.

Buona Idea!

DATA I/O

Programming systems for tomorrow.....
.....today.

SISTREL
SOCIETÀ ITALIANA STRUMENTI ELETTRONICI SpA

Via Pelizza da Volpedo 59 - 20092 CINISELLO B.
tel. (02) 6181893 - Telex 320346
Via Giuseppe Armellini 39, 00143 ROMA. Tel. (06) 5915553 - Telex 680356
Via Cintia Parco S. Paolo 35, 80126 NAPOLI - Tel. (081) 7679700

☐ Ricevere un'offerta ☐ La visita di un Vs. Tecnico ☐ Essere inseriti nel Vs. mailing list.

NOME .... COGNOME ....
VIA .... CITTÀ ....
CAP .... TEL. ....
DITTA ....
REPARTO ....

System 19

COMPUTER PRATICO accoglie idee e suggerimenti di utilizzo immediato nell'ambito dell'elaborazione personale. Lo scambio di quanto si è in grado di individuare lavorando con un computer è fondamentale per una diffusione effettiva dell'elaborazione personale. Per promuovere questa iniziativa, ogni testo pubblicato sarà ricompensato da un minimo di L. 50.000 ad un massimo di L. 100.000. I testi devono essere inviati a: Bit Rubrica COMPUTER PRATICO - Piazzale Massari, 22 - 20125 MILANO

Come "raddoppiare" un dischetto a singola faccia

di E. Macchi

Se avete un floppy-disk o un mini floppy-disk con una singola testina di registrazione, dovete adoperare dischetti a singola faccia. Ma una volta registrato l'intero dischetto non occorre metterlo da parte (e catalogarlo) fintantochè non lo avete "sfruttato" completamente. In che modo? Semplice: *registrando* sull'altra faccia.

Forse non sapete che entrambe le facce di un dischetto sono trattate per la registrazione e che l'inibizione alla registrazione sull'altro lato è realizzata unicamente dalla *custodia*. Occorre comunque un'avvertenza: dato che il costruttore certifica il dischetto come a singola faccia, la registrazione sul lato "vietato" potrebbe non essere sicura (ad esempio la caratteristica di MTBF —*M*ean *T*ime *B*etween *F*ailures— normalmente maggiore di 4000 ore, potrebbe essere inferiore).

Ciò premesso, vediamo in che modo si può registrare sulla faccia vietata del dischetto. Occorre distinguere tra mini-floppy e floppy (v. Figura) ed in particolare notare come sono disposti la *tacca di protezione da scrittura* e *il foro di indice*.

Mentre il rilevamento della tacca di protezione da scrittura è opzionale (nel senso che non tutti i driver hanno l'elettronica di rilevamento di questa tacca che, ricordiamolo, impedisce di scrivere sul dischetto stesso quando non è coperta), è invece fondamentale il rilevamento del foro indice che segnala al driver l'inizio del primo settore di ogni traccia. Infatti mentre la tacca di protezione da scrittura è fatta unicamente sulla custodia, il foro di indice è praticato sul dischetto di modo che quando, a causa della rotazione del dischetto, arriva all'altezza dei corrispondenti fori della custodia, è rilevato dai circuiti optoelettronici del driver che a questo punto può sincronizzarsi sull'inizio settore. Ora se introduciamo il dischetto nel driver nel solito modo ma con il lato vietato a contatto della testina di registrazione, notiamo che non vi sono problemi per la posizione della pista-testina, mentre la tacca di protezione da scrittura ma soprattutto il foro indice risultano essere collocati nella posizione simmetrica e quindi non possono essere rilevati dall'elettronica del driver.

È chiaro a questo punto quello che occorre fare: bisogna praticare sulla custodia i fori opportuni.

Attenzione: questa semplice operazione deve essere fatta con estrema cura e cioè bisogna stare attenti a non scalfire o comunque rovinare la superfice magnetica del dischetto e, una volta fatti i fori, bisogna badare a non lasciare sporca la superfice stessa o ancora a non lasciare, in corrispondenza del taglio, degli spuntoni che possano strisciare sul dischetto. Questo è tutto. Per finire incollate un'etichetta sul nuovo lato della custodia: ora avete un nuovo dischetto pronto all'uso. ■

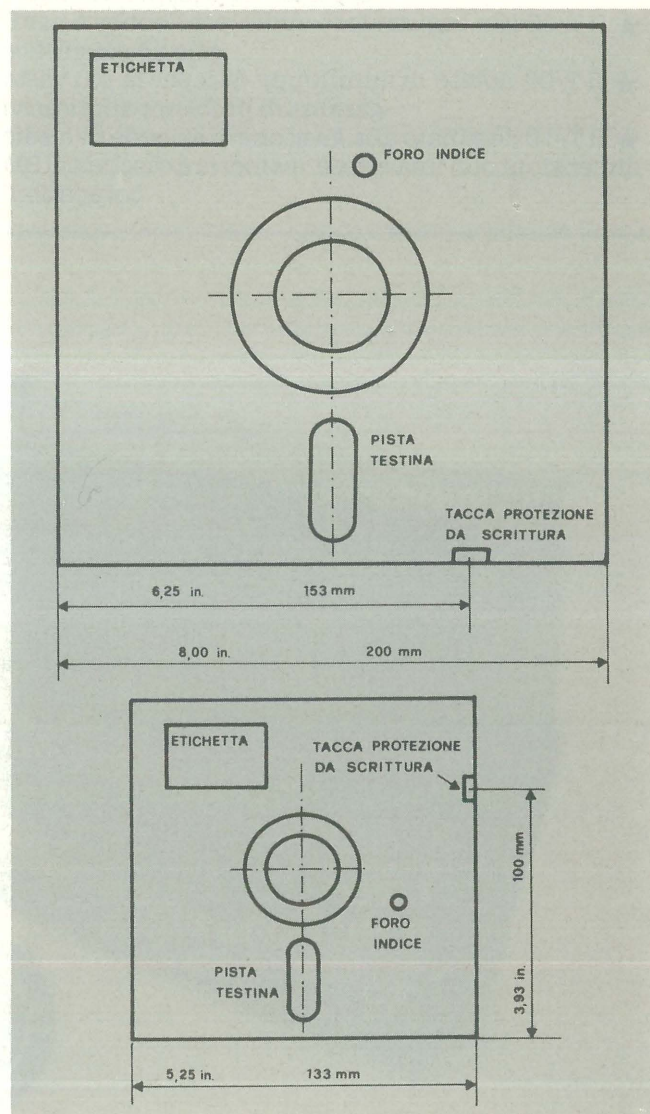



Figura 1 -

PERCHÈ ANCHE IL PIÙ ESIGENTE NON HA DUBBI PER SCEGLIERE GENERAL PROCESSOR?

Perché la GP ha più esperienza. La GP è la **prima** azienda italiana ad aver prodotto microcomputers e personal computers; la prima in ordine cronologico e la prima per produttività. È anche la prima per la sua rapida espansione.

Perché la diffusione dei prodotti GP è conferma di qualità. I sistemi GP entrano anche "negli ambienti che contano". Esperti tecnici, istituti universitari, industrie, enti di ricerca (come il Consiglio Nazionale delle Ricerche) si affidano ogni giorno al nome GP

Perché i prodotti GP sono i più prestigiosi. Ogni progetto è fatto con in mente l'utente finale, i suoi problemi, le sue esigenze. Nessun dettaglio è trascurato e la scelta dei componenti è fatta in base a criteri estremamente rigorosi.

Perché la gamma dei prodotti GP è estremamente vasta: il nuovo Modello T è completamente espandibile in senso verticale:

★ Il T/05 con registratore audio, per l'hobby o per il calcolo scientifico

★ Il T/08 dotato di minifloppy disk per la più vasta gamma di problemi applicativi

★ Il T/10 destinato alla gestione di aziende di medie dimensioni con una estesa memoria a dischetti IBM compatibili

★ Il T/20 con un grande disco da 14 a 24 Mega bytes che vi aspettereste di trovare solo su un sistema di costo molto maggiore.

Perché i «personal» della GP dispongono di una delle più vaste biblioteche software del mondo: Il nuovo Modello T è compatibile col famosissimo CP/M (*), il più diffuso sistema operativo a dischi oggi esistente. Sotto il CP/M (*) sono disponibili tutti i più conosciuti linguaggi di programmazione; quindi non più soltanto il BASIC, ma anche FORTRAN, COBOL, APL, PASCAL, BASEX, ASSEMBLER ecc. ecc. Il servizio software della GP è poi a vostra disposizione per personalizzare secondo le vostre necessità i numerosi programmi applicativi già realizzati o per studiarne dei nuovi. Problemi già risolti includono la contabilità generale, la gestione del magazzino, la contabilità semplificata, la prenotazione elettronica degli appuntamenti, il listino prezzi on line...

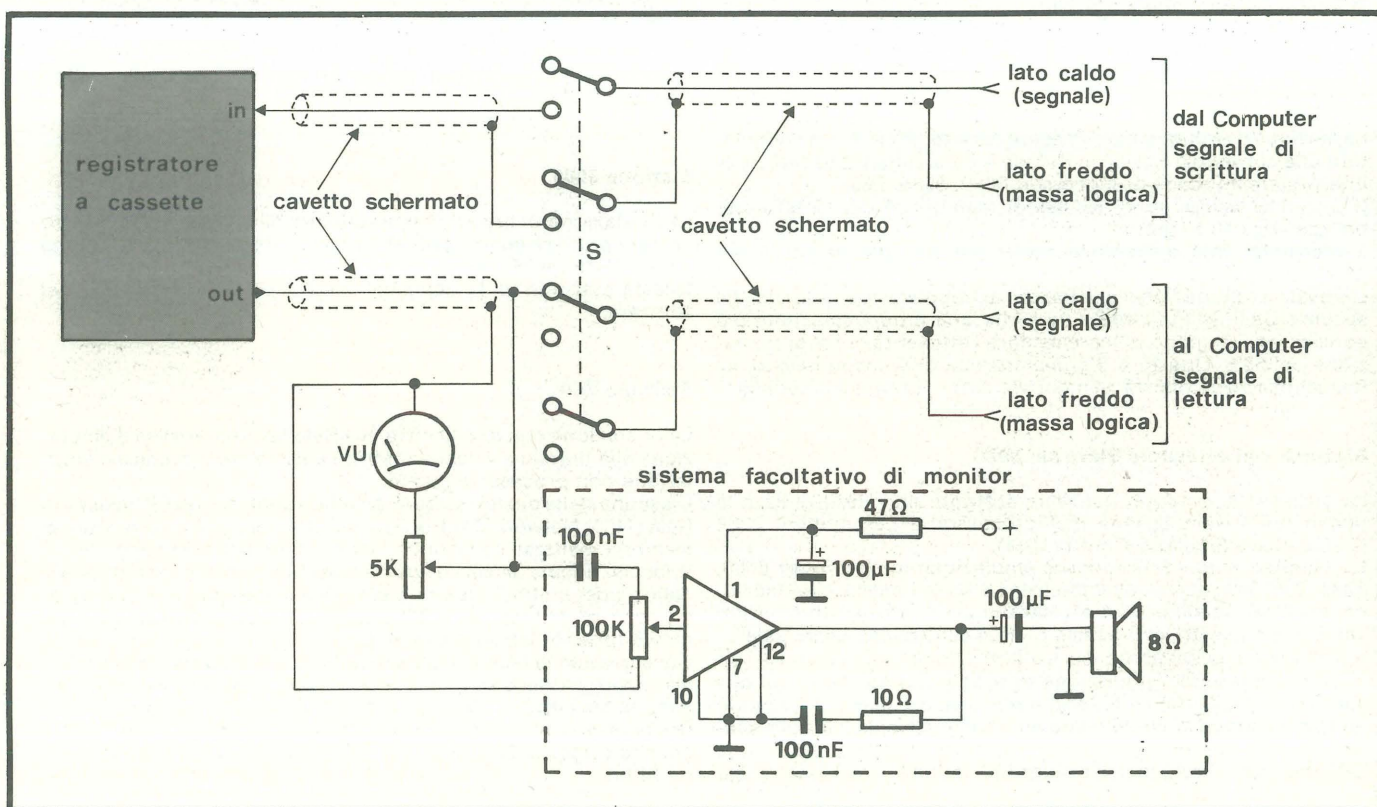
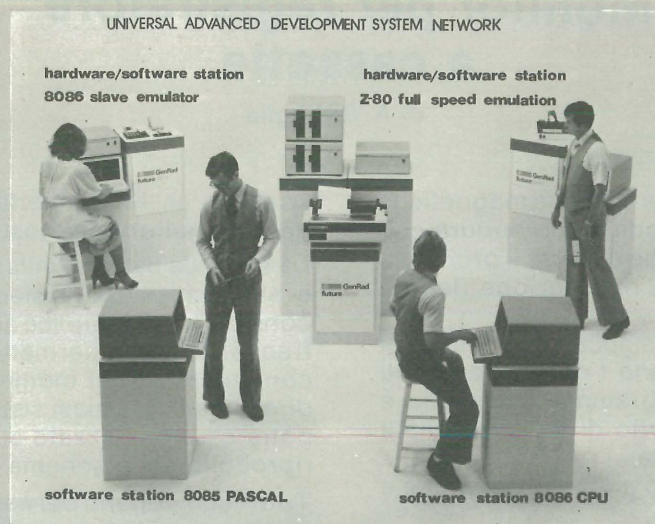
Perché l'assistenza di una ditta che opera in Italia è per forza la migliore. Una garanzia che solo una ditta italiana può offrire: la certezza di una buona e completa assistenza.

Qualunque sia il problema la risposta è una sola: General Processor. La General Processor è vicina; telefona (al mattino) allo 0.55 - 21.91.43.

(*) trade mark of Digital Research, USA

Monitor per registratore a cassette

di **A. Battaglia**

Quando si usa un registratore a cassette magnetiche in coppia con un sistema autodidattico oppure con un personal computer, potrebbe esistere il problema di tarare ogni volta il livello di riproduzione (lettura per il computer).
Inoltre può essere necessario staccare lo spinotto di registrazione quando si leggano i dati, e parimenti quando si deve salvare un programma su cassetta, è necessario reinserire lo spinotto di registrazione e staccare quello di riproduzione.
Questa serie di necessità sono alquanto scomode e tediose.
Per questa ragione propongo, lo schema elettrico di una semplicissima realizzazione che ci permette di ovviare a questo tipo di problemi. La semplicità dello schema non richiede molte spiegazioni, e la realizzazione potrà essere eseguita anche da chi di cablaggi elettronici non ha grande esperienza.
Il circuito riportato in Figura è costituito da due parti ben distinte.
La prima parte consiste basilarmente in un commutatore che abilita o disabilita l'I/O del registratore a cassetta. Con un unico comando quindi si potrà disinserire la registrazione ed abilitare la riproduzione, oppure al contrario sarà possibile abilitare la registrazione e sconnettere la riproduzione. In questa prima parte troviamo anche uno strumentino chiamato VU-meter che opportunamente tarato, *una-tantum* tramite il trimmer potenziometrico da 5000 Ω, ci permetterà di regolare il livello di uscita del registratore; un livello perfetto quindi secondo le esigenze di lettura del nostro computer.
La seconda parte, quella racchiusa dal tratteggio che è stata chiamata "sistema facoltativo di monitor", consiste in un semplice amplificatore lineare di bassa frequenza che ci permette l'ascolto del segnale che il computer in quel momento sta leggendo. È inutile dire che un qualsiasi sistema di amplificazione di BF potrà essere utilizzato al posto di quello proposto e riprodotto nello schema.
Tutti i componenti del nostro Monitor non sono assolutamente critici: in particolare l'integrato lineare di amplificazione di BF è un comune LM380.
Un contenitore metallico opportunamente forato sarà la giusta sede del "Monitor". Sul pannello frontale di questa scatola dovrà essere posto il VU-meter, e la manopola del commutatore; se fosse previsto il sistema di amplificazione si dovrà aggiungere un'interrutore di accesso spento ed una ulteriore manopola per la regolazione del volume. Sempre per chi volesse inserire la BF si potrebbe aggiungere una lampadina spia e sul retro si eseguiranno numerosi forellini in corrispondenza dell'altoparlante. L'alimentazione dell'amplificatorino di bassa frequenza potrà essere costituita da batterie a secco oppure da un alimentatore a bassa tensione opportunamente costruito od acquistato.
Una bella scatolina quindi, un pò di fori, e saldature *ben fatte* ed ecco uno strumentino di sicura utilità.

RETE DI SISTEMI DI SVILUPPO UNIVERSALI DELLA GENRAD

FUTUREDATA

Rete di stazioni di sviluppo universali per utenze multiple per sviluppo software ed hardware con processori da 8 a 16 bits.
La rete di sviluppo della GenRad Futuredata supporta fino ad 8 stazioni, ciscuna delle quali opera simultaneamente permettendo lo sviluppo di processori di natura diversa, linguaggio Assembly o ad alto livello (BASIC, Pascal).
Tramite il collegamento ad una unità di controllo, come ad asempio nella foto, una stazione utilizza una versione del Pascal per la serie 8080, un'altra supporta l'8086 con l'uso della unità d'emulazione Slave contenente l'analizzatore di stati logici e il modulo personalizzato con CPU 8086.
La terza stazione è impiegata per lo sviluppo del solo software con relativa esecuzione ma con una CPU interna da 16 bit 8086, mentre la quarta stazione opera lo sviluppo e l'emulazione in - circuit con analisi logica alla massima velocità per lo Z80.

Stazione Pascal

La stazione di sviluppo per il Pascal è caratterizzata da un compilatore che converte il Pascal in P-code e da un interprete capace di interpretare il P-code ai processori 8080, 8085, Z80.
Il Compiler lavora su 48 Kbytes di memoria RAM; l'interpreter utilizza circa 10 Kbytes.
L'interpreter sarà disponibile anche per altri processori: 8086, Z8000 ecc.....
L'elevata strutturazione del Pascal accoppiata alla velocità del sistema GenRad Futuredata permette una programmazione più concisa ed è in grado di incrementare l'efficienza di programmazione del 50%. Questo è il primo sistema di sviluppo basato sul Pascal che viene offerto all'industria con consegna immediata.

Stazione con emulatore Slave per 8086

La stazione di sviluppo hardware/software per l'8080 utilizza la nuova unità d'emulazione multiprocessor a bus multipli 2302 (SECU-Slave Emulator Control Unit).
La GenRad Futuredata fornisce emulazione in-circuit per 8080, 8085, Z80, 6800 e 6802, utilizzando la CPU del sistema. L'emulazione dell'8086, dello Z8000 o del 68000 e dei nuovi microprocessori verrà implementata con questa unità d'emulazione universale.
Il sistema Slave elimina qualsiasi limitazione degli emulatori incircuit permettendo emulazione di processori da 8 a 16 bit alla massima velocità (fino a 10 Mhz) e senza stati d'attesa. Utilizzando unità di controllo multiple si può operare una emulazione simultanea multiprocessor. L'avvento di prodotti utilizzanti più di una CPU ha creato l'esigenza d'emulazione multipla che viene perciò soddisfatta con questa unità.
La filosofia modulare dell'emulatore Slave permette inoltre di espandere rapidamente il supporto per altri nuovi processori con un tempo di progettazione di 6 settimane offrendo un servizio di supporto quasi immedato ed economico.
L'unità SECU 2302 è uno châssis separato in grado di recepire schede di memoria RAM statica fino a 128 Kbytes o 512 Kbytes di RAM dinamica per simulare le ROM/PROM. La memoria statica e la struttura del bus permettono un'emulazione fino a 10 Mhz.
L'unità incorpora inoltre un analizzatore di stati logici che fornisce 3 breakpoints ed un qualificatore di tracce da 48 bit. Il trace della memoria è di 64 bits x 256 eventi.
Packages di supporto aggiuntivi oltre all'8086 sono già disponibili per lo Z8000 e presto lo saranno per il 68000; consistono del relativo software e di una sola scheda personalizzata contenente la CPU in oggetto.
Entro il 1980 saranno disponibili anche i supporti per il 6502, il 6809 ed altri ancora.

Stazione 8086

La III stazione utilizza come processore di sistema un 8086 ed è uno dei primi sistemi di sviluppo universali con CPU da 16 bit che viene offerto all'industria.
Questa stazione ha la capacità di sviluppo e di esecuzione del software.

Stazione Z80

La IV stazione con lo Z80 offre alla Rete la caratteristica d'emulazione alla massima velocità (4MHz.) utilizzando la tecnica d'emulazione con processore ospite.
Ciascuna delle quattro stazioni tramite il controllore di Processo di Rete (NCP Network Control Processor) accede ai 4 Megabytes di memoria realizzati con 2 unità, doppio drive a dischi floppy, singola faccia, doppia densità e alla stampante ad alta velocità. Suddividendo così unità di memoria di massa e stampante, il costo per stazione di sviluppo si riduce drasticamente offrendo ad ogni utente tutte le risorse disponibili.
Conservando la filosofia della flessibilità, modularità e compatibilità, sono anche disponibili Drive a doppia faccia; verrà inoltre annunciata entro il 1980 una unità a dischi rigidi in grado di offrire una enorme capacità di memoria di massa tale da soddisfare le più critiche esigenze di una rete ad 8 terminali sviluppanti processori da 16 bit. ■

Sort in Basic

di **Roberto Bel**

Nell'ambiente di programmazione i terminali quali Sort e Sortizzare vengono usati, si può ben dire, quotidianamente. A cosa si riferisce questo tipo di sostantivo?
La risposta a questa domanda è abbastanza semplice, si può dire che il significato di Sort equivale ad ordinamento, estrazione, mettere in fila secondo una logica ben precisa, variare l'ordine dei records secondo una chiave, e così via su questo concetto base. Un programma di Sort veramente completo richiede molti K di memoria. Il nostro scopo però è quello di suggerire delle routines di ordinamento di records che al tempo stesso siano semplici e che occupino poco spazio in un programma.
È bene precisare, anche se ciò può essere intuitivo, che i programmi e le routines di Sort, tipo quelle che presentiamo, sono fatte solo per archivi con organizzazione sequenziale. Vale a dire per files che contengono records senza un indice che possa dire in quale posizione del file stesso sia posto il singolo record. La routinette che presentiamo possono tranquillamente venir poste all'interno di un programma e richiamate con opportuni comandi di GOSUB.

SORT semplice, prima routine.

Vediamo innanzitutto, prima di prendere in esame questa prima routine, di spiegare le funzioni che prendono le varie variabii in essa contenuta e che verranno anche per gli altri Sort che troveremo:

S = spostamento. Se S risulta uguale a 0 nessuno spostamento è stato eseguito durante la fase appena eseguita.
I = indice del loop di Sort
A = tabella contenente i dati da ordinare. Se detti dati fossero di natura alfanumerica è chiaro che questa variabile dovrà chiamarsi A$.
C = variabile di comando. Utilizzata nelle routine degli spostamenti dei dati.
N = numero dei dati da ordinare.

Quest'ultimo valore potrà essere dato da parametro esterno tramite una INPUT-N, se si conoscesse già a priori il numero dei records che saranno sottoposti al Sort. In caso contrario con una pre-routine si leggeranno tali records, incrementando ad ogni lettura un contatore opportunamente predisposto. La Figura 1 ci propone come semplice esempio una routinetta adatta al caricamento dei dati nella tabella A e nello stesso tempo viene seguito il conteggio della quantità dei records: i records vengono letti da una periferica che potrà essere una unità a cassette magnetiche oppure un altro tipo di device.

```
**00 N=1
**05 OPEN 1,(NUMERO DEVICE),0
**10 INPUT# 1, A(N)
**30 REM TEST END-OF-FILE
**40 IF ST THEN **100
**50 LET N=N+1
**60 GOTO **10
**100 CLOSE 1
**110 REM CONTINUAZIONE ALLA
**120 REM ROUTINE DI   S O R T
```

Figura 1 - Semplice routine di caricamento di dati in tabella A e conteggio dei records. Gli asterischi dovranno essere modificati in funzione della posizione della routine nel programma utente.

```
**00 REM ROUTINE DI SORT
**05 NUMERO  1
**10 LET S= 1
**20 FOR I=1 TO N-1
**30 IF A(I) <= A(I+1) THEN **80
**40 LET C=A(I)
**50 LET A(I) = A(I+1)
**60 LET A(I+1) = C
**70 LET S = 1
**80 NEXT I
**90 IF S = 1  THEN **10
```

Figura 2 - Prima routine di Sort. Per gli asterischi vale quanto detto nella didascalia della figura precedente.

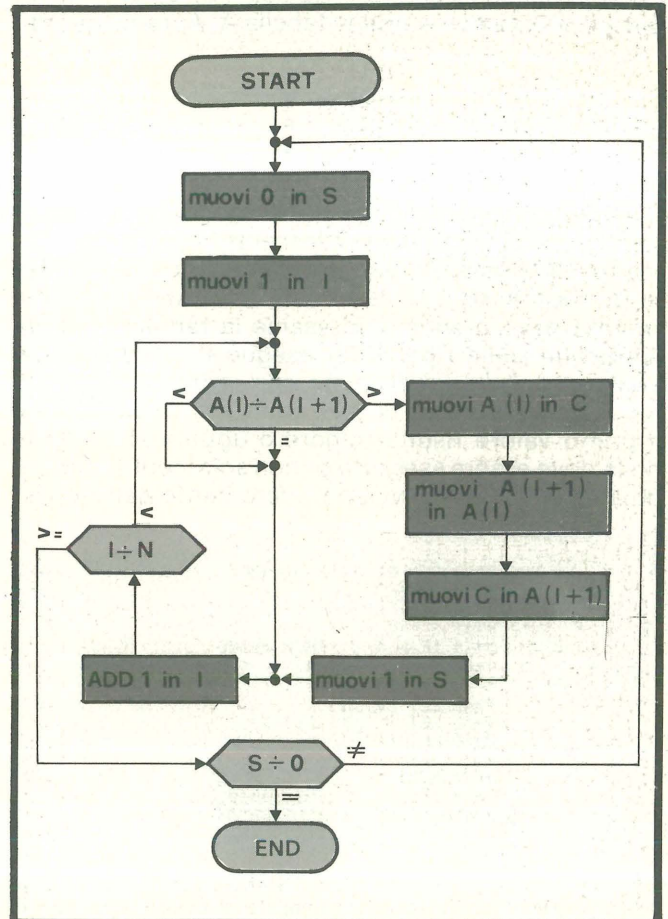



Figura 3 - Flow-chart della prima routine di Sort.

Elemento tabella	Contenuto iniziale	Loop 1			Loop 2			Loop 3			Loop 4		
		I = 1	I = 2	I = 3	I = 1	I = 2	I = 3	I = 1	I = 2	I = 3	I = 1	I = 2	I = 3
A (1)	10	9	9	9	8	8	8	7	7	7	7	7	7
A (2)	9	10	8	8	9	7	7	8	8	8	8	8	8
A (3)	8	8	10	7	7	9	9	9	9	9	9	9	9
A (4)	7	7	7	10	10	10	10	10	10	10	10	10	10
S =				1			1			1			0

Figura 4 - Tabella esplicativa dei vari passaggi e relativi spostamenti dei quattro records numerici presi come esempio, posti in ordine decrescente. Le freccette indicano gli spostamenti che il Sort decide di eseguire quando è soddisfatto il test di comparazione.

numero records da ordinare	totale comparazioni
4	12
6	30
8	56
10	90
20	380
50	2450
100	9900
255	64.770

Figura 5 - Tabellina di comparazione. Data la espressione $nC = N^2 - N$, si può determinare l'esatta quantità di comparazioni che il Sort deve eseguire.

Nella Figura 2 vediamo la routine di Sort vera e propria. Già con queste quattro semplici istruzioni è possibile eseguire un valido ordinamento di dati che in precedenza, come è stato detto, avevamo caricato in tabella A: [DIM A (numero di records)].
Che cosa accade in queste semplici istruzioni?
Prendiamo ad esempio i numeri 10, 9, 8, 7 ed in questo ordine poniamoli in tabella A. Avremo quindi:

$$A(1) = 10$$
$$A(2) = 9$$
$$A(3) = 8$$
$$A(4) = 7$$

L'obbiettivo del Sort sarà quello di restituire questi numeri nel giusto ordine crescente: 7, 8, 9, 10.
In *riga 10* poniamo a zero la variabile S che funzionerà da switch. In *riga 20* si definisce quante volte la routine deve prendere in esame la tabella A. Nella istruzione della *riga 30* si esegue il confronto del contenuto della tabella, ma con indice I+1, cioé il records seguente. Se questo confronto stabilisce che il primo valore risulta minore o uguale al secondo nulla deve essere eseguito se non solamente il rimando alla *riga 80* dove avviene l'incremento dell'indice I

```
**00 REM ROUTINE DI SORT
**05 NUMERO  2
**10 LET  S=0
**13 N = N - 1
**16 IF N = 0  THEN **100
**20 FOR I=1 TO N
**30 IF A(I) <= A(I+1) THEN **80
**40 LET C=A(I)
**50 LET A(I) = A(I+1)
**60 LET A(I+1) = C
**70 LET S = 1
**80 NEXT I
**90 IF S = 1  THEN **10
**100 REM FINE ROUTINE DI SORT
```

Figura 6 - Listing del secondo tipo di Sort. Routine estremamente simile a quella di Figura 2 ma ampliata di poche istruzioni: ci permette un minor numero di comparazioni.

e quindi si ritorna all'inizio del loop fino ad N operazioni.
Se il valore in esame risultasse maggiore del valore del seguente record ecco che si presenta la necessità di scambiare i dati presi in quel momento in comparazione. Per poter fare ciò in C, area di comodo, viene posto il record in esame definito in A(I), *riga 40*, quindi si sposta il dato posto in A(I+1) nel nuovo posto di tabella.
Se questi passi ora descritti sono stati eseguiti sarà necessario cambiare il valore di S affinché sia riconosciuto l'avvenuto scambio di posizione.
La routine di Sort viene eseguita fintantoché S viene modificato da valore zero a valore 1. Quindi è logico che quando troveremo S mantenuto a zero potremo dire che i records posti in tabella sono ordinati in modo crescente.
Osserviamo un attimo la tabella di Figura 4 dove vengono riportati i vari passaggi che la routine di Sort di Figura 2 esegue.
Già nella prima esecuzine del loop di Sort notiamo che al termine dei tre passi il numero 10 è già posto nella posizione esatta. Nel primo passo, cioé quando I=1, viene eseguito lo spostamento fra il numero 10 e il numero 9; nel secondo passo (I = 2) lo scambio di posto interessa i numeri 8 e 10; quando I sarà uguale a 3 lo scambio che avverrà interesserà sempre il numero 10 che in questa fase sarà comparato con il dato 7.
S è uguale ad 1 in quanto è stato eseguito come minimo uno scambio, quindi si riprende l'esecuzione del Sort.
Secondo giro: I è uguale ad 1 quindi lo scambio di posto che avviene è fra l'8 ed il 9; I = 2, scambio fra 7 e 9; I = 3 esame del 9 e del 10 già in ordine esatto quindi nessuno spostamento. S =.
Terzo loop di Sort: I = 1, esame di 8 con 7, spostamento; I = 2, comparazine fra 8 e 9, nessuno spostamento; I = 3, esame fra 9 e 10, nessuno spostamento. S = 1.

numero records da sortizzare	totale delle Sort n.1	comparazioni Sort n.2
4	12	10
6	30	21
8	56	36
10	90	55
20	380.	210
50	2.450	1.275
100	9.900	5.050
255	64.770	32.640

Figura 7 - Da questa tabellina esemplificativa si può dedurre, per confronto, il notevole risparmio delle comparazioni esguite dai Sort 1 e Sort 2.

Elemento tabella	Contenuto iniziale	Loop 1 FOR I = 1 TO N-1			Loop 2 FOR I = 1 TO N-2		Loop 3 FOR I = 1 TO N-3	N = 0
		I = 1	I = 2	I = 3	I = 1	I = 2	I = 1	
A (1)	10	9	9	9	8	8	7	
A (2)	9	10	8	8	9	7	8	
A (3)	8	8	10	7	7	9	9	End
A (4)	7	7	7	10	10	10	10	
S	0			1		1	1	0

Figura 8 - Tabellina dei vari passaggi eseguiti dalla seonda routine di Sort. Notare il notevole confronto con la tabellina di Figura 4.

Quarto giro: I = 1, test fra 7 e 8, nessuno spostamento; I = 2, test fra 8 e 9, nessuno spostamento; I = 3 esame fra 9 e 10, nessuno spostamento. S = 0. Fine esecuzione della routine di Sort.
A questo punto potremo dire che abbiamo eseguto quattro volte la nostra routine ed avremo eseguito ben 12 comparazioni.
La semplice tabellina di Figura 5 ci mostra alcuni esempi tratti dalla soluzione dell'espressione:

$$nC = N^2 - N$$

dove nC indica il numero totale delle comparazioni da eseguire per un numero N di records da sortizzare.

SORT numero 2

Se si volesse diminuire il numero di comparazioni necessarie per eseguire il richiesto ordinamento e quindi in conseguenza diminuire il tempo di esecuzione, si potrà aggiungere al precedente Sort due semplici istruzioni che fanno sì che il Sort non esegua delle comparazioni inutili. Aggiungiamo quindi le istruzioni di *riga 13* e *riga 16*, ed avremo trasformato la routine di Sort precedente descritta, in quella riportata nella Figura 6.
Dato che abbiamo tolto l'esecuzione di inutili comparazioni di records già posti in giusta allocazione di tabella, l'espressione che calcola le effettive comparazioni sarà:

$$nC = (N^2 + N) /2$$

La tabella precedentemente proposta subirà delle notevoli trasformazioni fino ad arrivare al cinquanta per cento delle comparazioni che in precedenza eravamo obbligati ad eseguire. Da notare anche che maggiore è il numero di records maggire è la percentuale di risparmio di comparazioni.
Senza doverci dilungare a descrivere i vari passaggi di questa nuova routine direi di osservare attentamente la tabellina di Figura 8 eventualmente confrontandola con la similare relativa al Sort descritto in precedenza.

```
**00 REM ROUTINE DI SORT
**05 NUMERO 3
**15 FOR I=1 TO N-1
**20 FOR K=I+1 TO N
**30 IF A(I) = A(K) THEN **70
**40 LET C = A(I)
**50 LET A(I) = A(K)
**60 LET A(K) = C
**70 NEXT K
**80 NEXT I
**90 REM FINE ROUTINE DI SORT
```

Figura 9 - Listing della terza routine di Sort. Nonostante il minor numero di istruzioni rispetto alle precedenti si può dire che fra le ruotine presentate questa risulta essere la migliore.

SORT numero 3

Un'altra interessante routine di Sort è riportata in Figura 9. Questo nuovo tipo di Sort ci permette di ottenere il medesimo risultato finale senza dover testare lo switch S degli scambi.
Ad ogni esecuzione di loop della routine ogni record viene comparato con tutti i suoi seguenti per N volte meno 1. Eliminando così l'elemento S si toglie anche le istruzioni relative al test di avvenuto scambio: ne consegue quindi una routine estremamente valida ed oltremodo semplice.
Quanto detto è dimostrato nella Figura 11, dove viene evidenziato il tipo di comparazione che viene eeguito di volta in volta. Minore occupazione delle istruzioni in memoria. Minori passi da eseguire. Minore tempo di esecuzione.

Breve conclusione

Già con questi tre tipi di routine di Sort sarà possibile avere ad esempio una ordinaria rubrica telefonia (or-

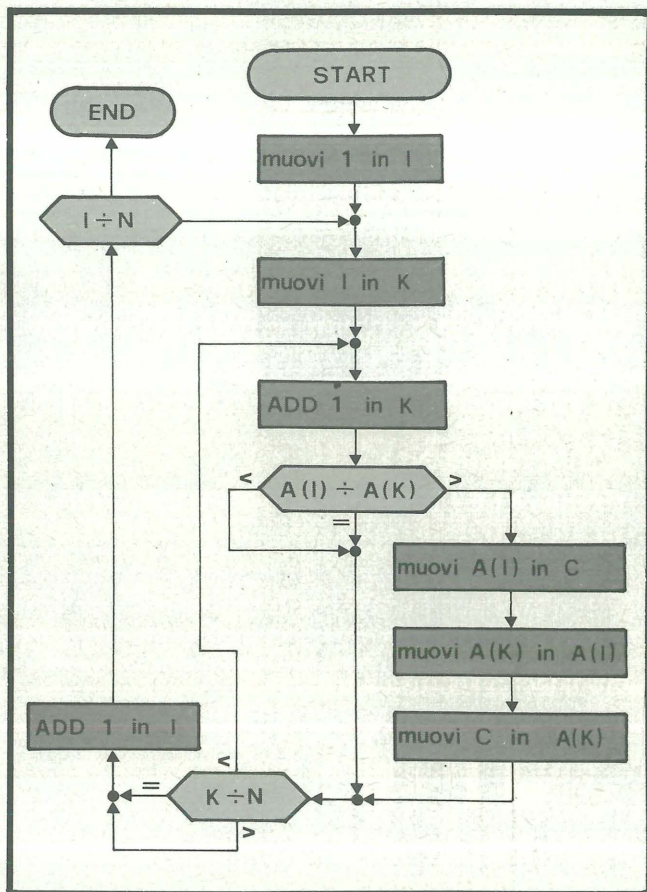
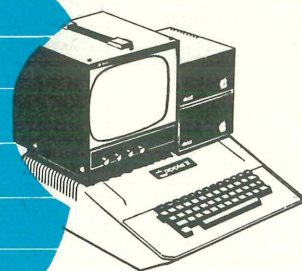



Figura 10 - Flow-chart della terza routine di Sort.

specialisti nel software per microprocessori e computer personali

FINALMENTE IN ITALIA IL PERSONAL COMPUTER PIU' VENDUTO NEL MONDO

apple II

SOFTEC, Agente esclusiva di vendita dell'Apple per la Lombardia, Piemonte e Liguria, vi offre per una completa utilizzazione dell'Apple II:

- seminari di introduzione ai Personal Computer
- corsi di Basic e Pascal
- seminari su sistemi operativi a dischi (DOS, UCSD)
- assistenza e consulenza sistemistica (pre e post-vendita)
- sviluppo di software applicativo per il mercato italiano (come il potente data base ARCHIVIO)
- disponibilità dei programmi generalizzati più diffusi nel mondo (VISICALC, EASYWRITER, DESKTOP PLAN, ecc.).

Il personal computer **PROFESSIONALE** dal costo più basso (oltre 100.000 Apple II venduti nel mondo sono un risultato eloquente!)

APPLE II E' PROFESSIONALE !

- Ha le caratteristiche costruttive e di qualità dei grandi calcolatori
- è dotato di software base completo e sofisticato (monitors, sistemi operativi, linguaggi, ecc.)
- è interfacciato a tutti i tipi di periferiche (stampanti, plotters, digitizer, comunicazioni, ecc.)
- è corredato di una completa documentazione tecnica e didattica
- è l'unico sistema che vi offre il "PLUS" della grafica (vera!) e del colore.

VOLETE SAPERNE DI PIU' ?
Telefonateci o veniteci a trovare.

Agenti Apple II per:
Lombardia, Piemonte e Liguria

MILANO - via G. Govone, 56 - tel. (02)3490231
TORINO - c.so M. d'Azeglio, 60 - tel. (011)6509303/4

CERCANSI aziende qualificate per la vendita di Personal Computers nelle zone ancora libere del Piemonte, Lombardia e Liguria.

Elemento tabella	Contenuto iniziale	Loop 1 I 1 K 2	K 3	K 4	Loop 2 I 2 K 3	K 4	Loop 3 I 3 K 4
A (1)	10	9	8	7	7	7	7
A (2)	9	10	10	10	9	8	8
A (3)	8	8	9	9	10	10	9
A (4)	7	7	7	8	8	9	10

Figura 11 - Terza tabellina dei passaggi. Da notare il diverso tipo di comparazione/spostamento eseguito dalla routine in esame, che è diverso dagli altri Sort (vedi articolo).

dinamento alfabetico) oppure un valido scadenziario (ordinamento numerico).
Mi sembra inutili rammentare che se i records presi in esame hanno un contenuto alfanumerico la DIM A(N) dovrà necessariamente essere cambiata con la relativa DIM A$(N) e quindi anche le istruzioni che contengono tale richiamo dovranno essere modificate di conseguenza.

La massiccia richiesta di software, venutasi a creare insieme con il grande successo dei "personal computers", ha dato lo spunto alla "bits & bytes" per creare una serie di programmi in linguaggio basic per il TRS - 80 della Radio Shack americana.
Questa azienda, la prima "tutta italiana" operante unicamente in questo settore, rappresenta un punto fermo nell'informatica, coprendo con i suoi programmi ogni campo, da quello gestionale a quello finanziario, da quello scientifico a quello ricreativo.
Trovando una valida risposta nel mercato italiano, la "bits & bytes", seppure di nuova costituzione, ha potuto arricchire il suo bagaglio di esperienze, offrendo un'alta professionalità e rispondenza alle reali esigenze di coloro che hanno già avuto l'opportunità di conoscerla ed apprezzarla.
Inoltre l'ottima qualità dei materiali e la cura con la quale essa opera, non trovano riscontro in nessuna altra società del settore. È interessante notare che per ciascun programma, l'incisione su nastri magnetici viene effettuata sempre più di una volta, avendo cura di utilizzare esclusivamente un lato del nastro, e ciò al fine di evitare possibili sovrapposizioni di segnali con le note disastrose conseguenze.
Va aggiunto inoltre che, per favorire la diffusione a tutti i livelli del "personal computer", la "bits & bytes" si adopera per mettere alla portata di qualsiasi operatore, seppure alle prime armi, tutti i programmi, anche i più sofisticati, in quanto essi, dotati di blocchi automatici, riducono al minimo l'eventualità di commettere errori.
In piena fase di espansione, la "bits & bytes" conta di sviluppare in tempi molto brevi una vastissima biblioteca di programmi e di soddisfare così ogni possibile richiesta della propria clientela.

ADESSO CHE COMPUTERIA E' A MILANO, VI BASTA POCO PER SAPERE COSA PUO' DARVI UNA VERA COMPUTERIA.

Computeria ha aperto a Milano un nuovo negozio: in via Moscova 24, angolo Corso di Porta Nuova.

Qui trovate l'elaboratore per tutte le esigenze e alla portata di tutti.

L'elaboratore per la gestione delle piccole aziende industriali, commerciali, di servizi.

L'elaboratore per chi ha uno studio professionale: ingegneri, commercialisti, architetti, analisti, ricercatori.

E, naturalmente, l'elaboratore per chi ama giocare con l'elaboratore, compreso il traduttore che vi traduce all'istante il vostro italiano in tutte le lingue.

Alla Computeria sono rappresentati i piú importanti costruttori del settore. Potete cosí finalmente rendervi conto della differenza tra una casa e l'altra, tra uno strumento e l'altro.

E scegliere quello che va bene per voi, con l'aiuto dei nostri tecnici.

Ma la Computeria non vi offre soltanto degli strumenti e la consulenza per aiutarvi nel loro acquisto. Vi offre un esperto servizio di assistenza tecnica sull'hardware e sul software applicativo, se e quando vi serve.

E vi dà tutto quello che vi occorre per lavorare col calcolatore: programmi applicativi, supporti magnetici.

Uno dei prossimi giorni, fate un salto alla Computeria: cosí, per curiosare.

Arrivederci presto.

COMPUTERIA®

Il Centro del Personal Computer

è marchio registrato della Unicomp S.r.l.

Computeria: 20121 Milano-Via della Moscova, 24-Tel. 02/666503

Unicomp: 20092 Cinisello Balsamo (Milano) palazzo Testi - Via Cantù, 20-Tel. 02/6121041

Un sistema didattico con lettore-perforatore Baudot

di **B. Fedel, M. Paoli** - *Gruppo di Studio sui Microprocessori, ENAIP, Trieste*

Premessa
Il nostro gruppo di studio ha iniziato ad occuparsi del problema microprocessori circa due anni fa. Questo tipo di scelta è stata da noi fatta per dare una risposta più concreta a tutta una serie di nuove problematiche che si pongono oggi nel mondo della formazione professionale, più specificatamente nel settore elettronico, e, più in generale, nella scuola.
I nostri obiettivi sono di vario tipo, anche se è possibile cogliere, ovviamente, un tema unificante:
- aggiornamento, riqualificazione e specializzazione di adulti, con particolare riguardo agli occupati nel settore delle aziende di non grandi dimensioni
- aggiornamento ricorrente dei docenti che operano nei Centri di formazione professionale e nel sistema scolastico in generale
- incremento della professionalità in seno al nostro Centro.

Risulta evidente, pertanto, come i nostri scopi siano prevalentemente didattici, nell'accezione più ampia che può essere data a tale termine.

Il sistema didattico DDCT F8

Il DDCT F8 System è controllato da un microprocessore F8 della Fairchild interfacciato con sei periferiche (v. Figura 1):
- una tastiera ASCII ed un terminale video; essi svolgono, particolarmente, funzioni di controllo al posto della TTY prevista dal programma di debugging (FAIR-BUG)
- un perforatore ed un lettore di nastro Baudot; vengono utilizzati, rispettivamente, per la perforazione e la memorizzazione in RAM di programmi oggetto, sorgenti, tabelle, ecc.
- una stampante ad aghi; l'uso è indirizzato prevalentemente alla stampa di listings e tabelle in genere
- un lettore di mini-cassette; viene utilizzato come memoria periferica di massa, sia di programmi che di dati.

L'F8 è un microprocessore ad 8 bit costituito fondamentalmente da una CPU, da una PSU e da una SMI o da una DMI. Nella CPU sono contenuti la ALU, l'accumulatore, lo Status Register, l'Instruction Register, il controllo di logica, il controllo di interrupt, il circuit di clock, 2 porte I/O ad 8 bit ciascuna e (particolarità estremamente interessante) 64 registri RAM ad 8 bit, chiamati Scratchpad. Tali registri sono indirizzati in modo indiretto mediante un registro a 6 bit (ISAR) il cui contenuto può essere incrementato, decrementato o lasciato inalterato dopo l'accesso al contenuto del registro. Per i primi 16 registri esiste la possibilità di indirizzamento diretto.

È evidente come lo Scratchpad rappresenti un'alternativa all'uso di stacks interni all'area di indirizzo di memoria. Nella PSU sono contenute 2 porte I/O ad 8 bit ciascuna, un controllo di interrupt, un local timer, un Data Counter, due Program Counters ed una ROM da 1 Kbytes che contiene il programma residente FAIR-BUG. Nella 3851 APC esso occupa le posizioni di memoria da 8000 a 83FF e gestisce lo scambio di informazioni tra il microprocessore ed il mondo esterno non con una serie di displays (come accade per la maggior parte dei sistemi di valutazione) bensì direttamente su TTY. Come già detto, noi abbiamo sostituito alla telescrivente la tastiera ASCII ed un CRT mediante un modulo di interfaccia video. Il FAIR-BUG consente pertanto un colloquio più naturale col microprocessore rispetto agli altri sistemi di valutazione. I programmi di input/output sono organizzati come subroutines direttamente richiamabili dall'operatore nei propri programmi. Il FAIR-BUG contiene inoltre un programma per lettore e perforatore parallelo ASCII (LOADER FORMAT).
La SMI decodifica l'indirizzo per memorie statiche fino a 64 Kbytes; contiene due Data Counters, due Program Counters, una logica di interrupt ed un local timer indirizzabile.
La DMI decodifica l'indirizzo per le memorie dinamiche; contiene la logica di rinfresco, due Program Counters e due Data Counters.
Il microprocessore di cui noi attualmente disponiamo è basato sui primi tre chips.

Il microprocessore è stato da noi acquisito circa due anni fa, sotto forma di Evaluation Card. Unica periferica che avemmo immediatamente a disposizione, fu una TTY Baudot della Olivetti: il che ci costrinse ad

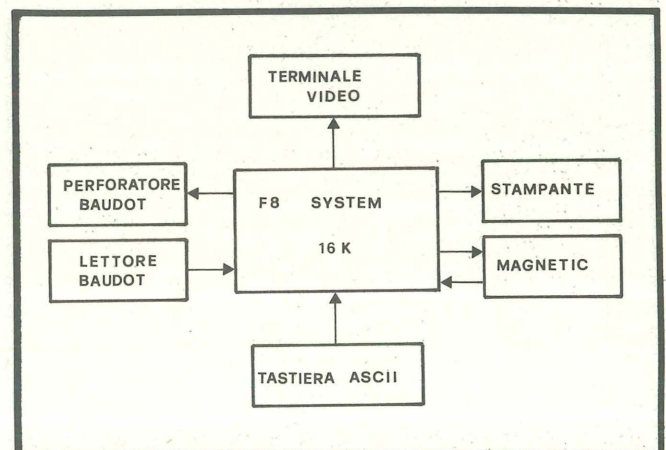



Figura 1 - Schema a blocchi del DDCT F8 System.

```
                              DDCT F8 SYSTEM

0001  *
0002  *THIS SUBROUTINE TRANSLATES BAUDOT CHARACTERS INTO ASCII
0003  *CHARACTERS
0004  *BAUDOT CHARACTERS ARE RECEIVED BY I/O PORT 0 BIT 7
0005  *
0006        ORG  H'0000'
0007  BAUD  LR   K,P
0008        LR   A,KU
0009        LR   QU,A
0010        LR   A,KL
0011        LR   QL,A      RETURN ADDRESS SAVED INTO Q REGISTER
0012        XDC
0013  T100  PI   CRDI      READING OF FIRST CHARACTER. IF THIS IS
0014        CI   1F        FIGURE OR LETTER, THE STATE IS SAVED INTO
0015        BNZ  T101      REGISTER 6
0016  T102  LR   6,A
0017        BR   T100
0018  T101  CI   1B
0019        BNZ  T103
0020        BR   T102
0021  T103  LR   A,6       SELECTION OF STACK MEMORY FIGURE OR LETTER
0022        CI   1F
0023        BZ   L1
0024        DCI  FGR
0025        BR   L2
0026  L1    DCI  LTR
0027  L2    LR   A,1
0028        ADC            THE CHARACTER IS LOADED INTO DC0 REGISTER
0029        LM             ASCII TRANSLATION IS LOADED INTO ACCUMULATOR
0030        LR   1,A
0031        XDC
0032        LR   P0,Q      END OF TRANSLATION SUBROUTINE
0033  *
0034  *SUBROUTINE INPUT BAUDOT CHARACTER
0035  *
0036  CRDI  LR   K,P       RETURN ADDRESS SAVED INTO K REGISTER
0037  C2    LIS  1
0038        LR   1,A
0039        OUTS 0
0040        LIS  6         THE NUMBER OF RECEIVED BITS IS SAVED INTO
0041        LR   2,A       REGISTER 2
0042  C1    INS  0         WAIT START BIT
0043        BM   C1
0044        PI   WAIT
0045        INS  0         VERIFY START BIT
0046        BM   C2
0047  C4    PI   WAIT
0048        PI   WAIT
0049        INS  0
0050        NI   [illegible]
0051        AS   1
0052        DS   2
0053        BZ   C3
0054        SR   1
0055        LR   1,A
0056        BR   C4
0057  C3    SR   1
0058        SR   1
0059        NI   1F
0060        LR   1,A
0061        PK
0062  *
0063  *SUBROUTINE 11 MS WAIT
0064  *
0065  WAIT  LI   D0
0066        LR   5,A
0067  B1    OUTS 0
0068        OUTS 0
0069        OUTS 0
0070        OUTS 0
0071        OUTS 0
0072        DS   5
0073        BNZ  B1
0074        POP
0075  *
0076  *
0077        ORG  H'005D'
0078  FGR   DC   H'0033'   FIGURE BAUDOT CHARACTERS  STACK MEMORY
0079        DC   H'0A0D'
0080        DC   H'2027'
0081        DC   H'3037'
0082        DC   H'0D21'
0083        DC   H'347F'
0084        DC   H'2C2A'
0085        DC   H'3A2C'
0086        DC   H'3530'
0087        DC   H'2032'
0088        DC   H'2230'
0089        DC   H'3431'
0090        DC   [illegible]
0091        DC   H'2A28'
0092        DC   H'2E2F'
0093        DC   H'3D'
0094  *
0095  *
0096        ORG  H'007A'
0097  LTR   DC   H'0045'   LETTER BAUDOT CHARACTERS  STACK MEMORY
0098        DC   H'0A41'
0099        DC   H'2053'
0100        DC   H'4959'
0101        DC   H'0D44'
0102        DC   H'5248'
0103        DC   H'4E43'
0104        DC   H'4A4D'
0105        DC   H'5450'
0106        DC   H'4C57'
0107        DC   H'4850'
0108        DC   H'5851'
0109        DC   H'4F42'
0110        DC   H'4700'
0111        DC   H'4D58'
0112        DC   H'56'
0113        END
```

Figura 2 - Listing del traduttore Baudot in ASCII.

orientare i nostri sforzi iniziali verso la costruzione di un traduttore hardware ASCII-Baudot e viceversa. Dopo un primo periodo di assuefazione al microprocessore, siamo passati all'autocostruzione di una espansione con completo isolamento di tutte le porte di I/O, simulatori di entrata e uscita. Vi sono inoltre una RAM di 17 Kbytes ed una ROM di 5 Kbytes. Effettivamente, la scelta iniziale di tale tipo di microprocessore è stata in parte casuale, in parte dovuta ad una sicura reperibilità da parte del nostro fornitore. A questo punto, però, ci rendiamo conto che dal punto di vista didattico, rispetto ad altri microprocessori, offre una contemporanea semplicità e completezza del set di istruzioni, un interessante programma di debugging, un modo di indirizzamento vario ma non complesso. Infine, essendo un microprocessore poco noto, almeno nel campo dei personal computers, *costringe* l'utente a crearsi un software di base accompagnato dal relativo hardware, il che rappresenta un grosso stimolo da cui deriva un aggiornamento continuo e costante.

Utilizzo della TTY Baudot

Una volta acquisite le periferiche già citate, ci si è posto il problema di utilizzare il lettore-perforatore (TTY) Baudot, come d'altra parte la necessità di una gestione di una memoria periferica non complicata, come può essere quella di un lettore di cassette con ricerca veloce.
Abbiamo escluso l'utilizzo del traduttore hardware, precedentemente realizzato, in quanto imponeva un rallentamento del clock da 2 MHz a circa 700 KHz. Del resto l'acquisizione di lettori-perforatori ASCII di nastro comportava per noi problemi economici non indifferenti. Abbiamo così realizzato due subroutines: la prima (BAUD) permette di tradurre un carattere Baudot seriale, ricevuto attraverso il bit 7 della porta 0, in carattere ASCII; la seconda (ASCI) traduce un carattere ASCII in Baudot, trasmesso serialmente attraverso il bit 0 della porta 0.

Conversione software del codice Baudot in ASCII

La BAUD (v. Figura 2) richiama inizialmente una subroutine (CRDI) che pone in accumulatore e nel registro 1 dello Scratchpad il carattere Baudot ricevuto serialmente alla velocità di 45,45 baud. Per il corretto funzionamento della subroutine va notato, per inciso, che il primo carattere presente sul nastro perforato deve essere necessariamente il codice LETTERE o CIFRE.
Il dato così ricevuto viene campionato con il codice LETTERE o CIFRE: se tale campionamento è positivo, lo stato viene memorizzato nel registro 6 e di conseguenza viene indirizzato lo stack memory che contiene l'elenco della traduzione dei caratteri. All'indirizzo così determinato dello stack memory viene sommato il dato ricevuto; la locazione di memoria così indirizzata contiene la traduzione del codice.
La subroutine CRDI attende il bit di partenza e lo campiona dopo un tempo pari alla metà della sua durata. Ciò per evitare che impulsi spuri e di breve durata presenti sulla porta siano interpretati come bit di partenza. Essa richiama una ulteriore subroutine (WAIT) che ha il compito di fornire un tempo di attesa di 11 ms, tenendo conto di un clock di 2 MHz.
In tale traduzione alcuni caratteri Baudot non sono

BAUDOT	ASCII
%	$
BELL	RUBOUT
DEGREE	*
CHI E'?	#
;	"

Figura 3 - Tabella dei simboli con traduzione particolare.

stati tradotti nei corrispettivi ASCII: ciò è stato fatto per le esigenze del sistema operativo, che utilizza simboli particolari che non sono presenti sulla tastiera della TTY (vedi tabella di Figura 3).

Conversione software del codice ASCII in Baudot

La seconda subroutine, come già detto, traduce i caratteri ASCII in Baudot. La struttura fondamentale è simile a quella della subroutine precedente: cioè il codice del carattere viene sommato al DC1 ed il nuovo indirizzo contiene la traduzione cercata.
Per usare correttamente la subroutine è necessario che la posizione del carrello della TTY sia compatibile con il contenuto del registro 6; cioè che quando esso contiene 1 il carrello abbia la posizione *upper* e quando contiene 0 abbia la posizione *lower*. Per fare ciò è necessario che, all'atto del primo accesso alla subroutine, si emetta il carattere LETTERE o CIFRE e si carichi nel registro 6, 1 o 0 rispettivamente. Ciò non è stato incluso nella subroutine ASCI in quanto, in tale ipotesi, verrebbe emesso per ogni carattere anche il corrispondente stato; l'uso dell'ASCI così organizzata, invece, emette il carattere di stato solo quando esso varia, con conseguente guadagno di tempo.
Poiché lo stack memory è organizzato in maniera da porre in esso i caratteri Baudot secondo il codice ASCII crescente, per evitare un'eccessiva lunghezza la parte iniziale della subroutine esegue un campionamento in modo da individuare la presenza di uno dei quattro caratteri con codice molto diverso dagli altri, e cioè *Carriage Return, Line Feed, Blank, Null*; a questi codici viene sommato un valore opportuno in modo da rientrare in coda allo stack.
Come si nota dal listing (v. Figura 4) la CRDO, richiamata dall'ASCI, emette un bit e mezzo di stop e utilizza come timer della durata di un bit la stessa subroutine WAIT utilizzata nella BAUD.
Per quel che riguarda la presenza di caratteri particolari non presenti nel codice Baudot (o, comunque, sulla tastiera della TTY), vale il discorso fatto a proposito della BAUD, eccezion fatta per il RUBOUT, che non viene tradotto.

Riportiamo, per completare l'argomento, una tabella (v. Figura 5) nella quale sono riportati i codici ASCII e Baudot. Essa è ovviamente comprensiva della Tabella 3.

Per quanto concerne il codice Baudot, ricordiamo che esso è a 5 bit (per cui può rappresentare al massimo 32 caratteri per ogni posizione del carrello) preceduti da un bit di partenza e seguiti da un bit e mezzo di stop.
La velocità è di 45,45 baud (22 ms per bit) oppure di 50 baud (20 ms per bit) a seconda del tipo di telescrivente. Alla velocità di 45,45 baud un byte dura allora 165 ms.

```
                         DDCT F8 SYSTEM

0001  *
0002  *THIS SUBROUTINE TRANSLATES ASCII CHARACTERS INTO BAUDOT
0003  *CHARACTERS
0004  *THE BAUDOT CHARACTER AND HIS LETTERS' OR FIGURES' STATE
0005  *OUTPUT BY I/O PORT 0 BIT 0
0006  *
0007       ORG  H'0099'
0008  ASCI LR   K,P
0009       LR   A,KU
0010       LR   QU,A
0011       LR   A,KL
0012       LR   KL,A      RETURN ADDRESS SAVED INTO K REGISTER
0013       XDC            DC1 REGISTER IS THE COUNTER OF BAUDOT
0014       DCI  BDT       CHARACTER TABLE
0015       LR   A,1
0016       CI   20        BLANK TEST
0017       BNZ  F2        BRANCH IF NOT BLANK
0018       LI   37
0019       BR   F10
0020  F2   CI   0D        C/R TEST
0021       BNZ  F3        BRANCH IF NOT C/R
0022       LI   38
0023       BR   F10
0024  F3   CI   0A        LINE FEED TEST
0025       BNZ  F4        BRANCH IF NOT LINE FEED
0026       LI   39
0027       BR   F10
0028  F4   CI   00        NULL TEST
0029       BNZ  F11       BRANCH IF NOT NULL
0030       LI   3A
0031       BR   F10
0032  F11  LR   A,1
0033       AI   DF
0034  F10  ADC            INTO DC1 REGISTER BAUDOT CHARACTER ADDRESS
0035       LM             INTO ACCUMULATOR BAUDOT CHARACTER
0036       LR   7,A
0037       LR   A,1       RESEARCH LETTER OR FIGURE STATE
0038       AI   BF
0039       BM   F30
0040       LIS  1
0041       XS   6         PREVIOUS LETTER STATE TEST
0042       BZ   F40
0043       LIS  1
0044       LR   6,A       STATE SAVED INTO REGISTER 6
0045       LI   1F
0046  F50  LR   1,A
0047       PI   CRDO      EMISSION LETTER OR FIGURE CHARACTER
0048  F40  LR   A,7
0049       LR   1,A
0050       PI   CRDO      EMISSION BAUDOT CHARACTERS
0051       XDC
0052       LR   P0,Q      END OF SUBROUTINE
0053  F30  CLR
0054       XS   6         PREVIOUS FIGURE STATE TEST
0055       BZ   F40
0056       CLR
0057       LR   6,A
0058       LI   1B
0059       BR   F50
0060  *
0061  *SUBROUTINE OUTPUT BAUDOT CHARACTER
0062  *
0063  CRDO LR   K,P
0064       LIS  7
0065       LR   2,A
0066       CLR
0067       OUTS 0         START BIT EMISSION
0068  C2   PI   WAIT
0069       PI   WAIT
0070       DS   2         DECREMENT BIT COUNTER
0071       BNZ  C1
0072       LIS  1
0073       OUTS 0
0074       PI   WAIT      EMISSION HALF STOP BIT
0075       PK
0076  C1   LR   A,1
0077       NI   01
0078       OUTS 0
0079       LR   A,1
0080       SR   1
0081       AI   10
0082       LR   1,A
0083       BR   C2
0084  *
0085  *
0086       JRG  H'010A'
0087  BDT  DC   H'0014'   STACK MEMORY BAUDOT CHARACTERS
0088       DC   H'091A'
0089       DC   H'1A00'
0090       DC   H'050F'
0091       DC   H'120D'
0092       DC   H'110C'
0093       DC   H'031C'
0094       DC   H'1D16'
0095       DC   H'1713'
0096       DC   H'010A'
0097       DC   H'1015'
0098       DC   H'0706'
0099       DC   H'180E'
0100       DC   H'1400'
0101       DC   H'1E00'
0102       DC   H'1900'
0103       DC   H'0319'
0104       DC   H'0E09'
0105       DC   H'010D'
0106       DC   H'1A14'
0107       DC   H'0608'
0108       DC   H'0F12'
0109       DC   H'1C0C'
0110       DC   H'1316'
0111       DC   H'170A'
0112       DC   H'0510'
0113       DC   H'071E'
0114       DC   H'131D'
0115       DC   H'1511'
0116       DC   H'0408'
0117       DC   H'0200'
0118       END
```

Figura 4 - Listing del traduttore ASCII in Baudot.

ASCII CODICE	ASCII CAR.	BAUDOT CODICE FGR	BAUDOT CODICE LTR	BAUDOT CAR.
00	NULL	00	00	NULL
0A	LINE FEDD	02	02	LINE FEED
0D	RETURN	08	08	RETURN
20	BLANK	04	04	BLANK
22	"	14		;
23	#	09		CHI E'?
24	$	1A		%
27	'	05		'
28	(	0F		(
29	)	12		)
2A	*	0D		DEGREE
2B	+	11		+
2C	,	0C		,
2D	-	03		-
2E	.	1C		.
2F	/	1D		/
30	0	16		0
31	1	17		1
32	2	13		2
33	3	01		3
34	4	0A		4
35	5	10		5
36	6	15		6
37	7	07		7
38	8	06		8
39	9	18		9
3A	:	0E		:
3D	=	1E		=
3F	?	19		?
41	A		03	A
42	B		19	B
43	C		0E	C
44	D		09	D
45	E		01	E
46	F		0D	F
47	G		1A	G
48	H		14	H
49	I		06	I
4A	J		0B	J
4B	K		0F	K
4C	L		12	L
4D	M		1C	M
4E	N		0C	N
4F	O		18	O
50	P		16	P
51	Q		17	Q
52	R		0A	R
53	S		05	S
54	T		10	T
55	U		07	U
56	V		1E	V
57	W		13	W
58	X		1D	X
59	Y		15	Y
5A	Z		11	Z
7F	RUBOUT	0B		BELL

Figura 5 - Tabella dei caratteri Baudot con i corrispettivi ASCII insieme ai codici in notazione esadecimale.

Il codice ASCII normalmente usato è quello a 7 bit (per cui può rappresentare fino a 128 caratteri diversi) preceduti da un bit di partenza e seguiti da un bit di parità e da due bit di stop. Le velocità standard di trasmissione del byte ASCII sono: 110, 300, 1200 baud. La durata di un byte completo è pertanto, rispettivamente, di 100, 36,6 e 9,16 ms.
Il DDCT F8 System lavora a 300 baud con il CRT e la tastiera ASCII, a 45,45 baud con il lettore-perforatore di nastro, a 1200 baud con la stampante e a 2400 baud con il lettore di mini-cassette.
A proposito delle velocità di trasmissione ricordiamo che il FAIR-BUG dà la possibilità di scelta fra due velocità standard di 100 e 300 baud ed una velocità programmabile da 107,8 a 10989 baud con un clock di 2 MHz.

Conclusione

Ci rendiamo conto che può sembrare anacronistico continuare a parlare ancora oggi di periferiche Baudot, ma non bisogna dimenticare che queste si trovano sul mercato surplus a costi molto accessibili da parte di qualsiasi utente; mentre una telescrivente ASCII presenta attualmente costi che riteniamo proibitivi ai più. ■

Bibliografia

1) *F8 User's Guide* (Fairchild)
2) *F8 Guide to Programming* (Fairchild)
3) *F8 FAIR-BUG User's Guide* (Fairchild)

ACCESSO CASUALE

COMPONENTI

Microcomputer a 2 chips Motorola

La Motorola presenta un insieme di due chips, atti a formare un potente microcomputer. Il primo chip è il 6802 che è la versione migliorata del "vecchio" 6800, in quanto comprende anche il clock onboard e 128 bytes di RAM. Il 6802 è alimentato con una alimentazione unica di + 5V e si collega in questo caso col secondo chip, il 6846.
Questo comprende una ROM di 2 Kbyte, dieci linee di I/O, Timer-/counter, con tre linee di accesso e uscita.
La struttura presentata è un valido punto alternativo nei confronti dell'applicazione del 6801, microcomputer a single chip che riunisce le caratteristiche dei due chips presentati sopra in un unico integrato, fornibile con ROM o EPROM.
Dunque l'utente ha la possibilità di valutare queste due alternative, sia per quanto riguarda maggiori o minori difficoltà nell'applicazione e nell'uso, sia per quanto riguarda le possibilità di fornitura (e di prezzo) da parte della Motorola.
In ogni caso la struttura è perfettamente compatibile con tutte le periferiche della famiglia 6800.

MOTOROLA ITALIA - MILANO

Signal processor Intel 2920-10

L'Intel presenta un processore di segnale, il 2920.
Questo è un "single chip" programmabile con uscite di segnale analogico e digitale disegnato per rimpiazzare dei sottosistemi analogici in applicazioni di processing in tempo reale.
Il suo set di istruzioni, più la CPU digitale ad alta precisione (25 bit), consentono una capacità notevole di implementare sottosistemi molto complessi.
Le tipiche funzioni ottenibili col 2920 comprendono: filtri passa basso e passa banda fino a 20 coppie di poli/zeri; rivelatori di soglia, limitatori, rettificatori, divisioni e moltiplicazioni fino a 25 bit, approssimazioni di funzioni non lineari come la legge quadratica ed i logaritmi, operazioni logiche, input ed output con multiplex di segnali, output logici per decisioni di processo, output analogici per oscillatori multifrequenza, generatori di frequenza ecc.
Per di più, molti 2920 possono essere messi in cascata per applicazioni di processing molto complesse con nessuna perdita nel throughput.

ELEDRA - MILANO

EPROM Texas Instruments

La T.I. annuncia la nuova serie di memorie EPROM.
La famiglia di tali EPROM comprende la 2508 (2708 ad una sola alimentazione, 1K × 8); la 2516 (idem, 2K × 8); la 2532 (4K × 8) e la 2564 (8K × 8).
I tempi di accesso sono standard, 450 ns per i chips da 32 K e 64 K mentre esiste la possibilità di avere le 2508 / 2516 con tempi di accesso rispettivamente di 250 / 300 ns e 350 / 450 ns.
La comodità della singola alimentazione è indubbia; inoltre tale famiglia di EPROM si programma in un unico modo standard e molto semplice, in quanto al pin di PROGRAM non deve (come la vecchia 2708) essere applicato un impulso di 25 Volt, bensì è sufficente applicare un impulso TTL della durata di circa 50 ms (dopo che l'indirizzo del byte da programmare si è stabilizzato, come pure il dato da applicare alla cella stessa).
Le caratteristiche generali della famiglia sono un completo funzionamento statico, parola di 8 bits, chip-select / power down automatico e bassa dissipazione.
Si deve notare che la TMS 2564, pur essendo impaccata in un package a 28 pins è compatibile con le ROM da 64 K impaccate in 24 pin e con le EPROM più piccole.
Questo dipende dal fatto che i pins dal 3 al 26 della EPROM da 8K × 8 sono compatibili coi pins da 1 a 24 dei dispositivi a 24 pins. La compatibilità è raggiunta riservando sia il pin 28 che il pin 26 per l'alimentazione a 5 Volt. In questa maniera, portando l'alimentazione al pin 26, si possono utilizzare dispositivi sia a 24 che a 28 pins senza jumpers da cablare. Tutte le EPROM T.I. sono disponibili in packages da 600 mil. con pin-outs JEDEC standard.

TEXAS INSTRUMENTS CITTADUCALE (RIETI)

SISTEMI DI VALUTAZIONE

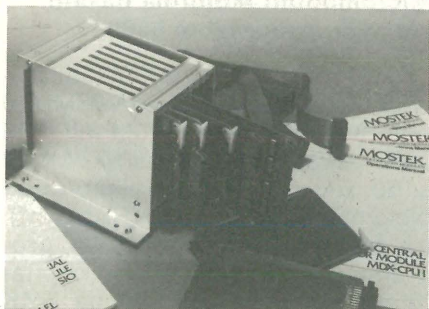
Micromodulo Motorola basato sul microprocessore 6809

La Motorola annuncia l'introduzione di un modulo chiamato M68MM19 per applicazioni industriali col 6809. Le caratteristiche dell'M68MM19 sono: I/O seriale e parallelo completamente bufferizzato; RAM (2K), tre funzioni timer / counter programmabili; zoccoli atti ad ospitare sia memorie EPROM da 2/4K

che RAM compatibili; espandibilità con altri moduli periferici e compatibilità con i BUS standard EXORciser e EXORterm.
Una analisi più dettagliata dell'hardware sulla scheda vede la presenza delle seguenti sezioni:
—MC6809 microprocessor + Quarzo a 4 MHz.
—Printer, Parallel I/O (PIA e Buffers).
—Timer I/O (Programmabile Timer Outputs/Inputs).
—RAM & ROM.
—Refresch DMA Control.
—Baud Rate Generator, ACIA, RS232C e RS422 adaptors, matrix, tutto ciò relativo alle linee del port di I/O seriale.
—Buffers sui Dati, Indirizzi e Linee di Controllo.
La struttura di tale modulo e la potenza del 6809 permettono una drastica riduzione dei tempi e dei modi di progetto nell'hardware.
Questo nuovo micromodulo può essere agevolmente inserito nel Mother dell'EXORciser, permettendo così all'utente di accedere alle enormi possibilità operative del microproccessore 6809. Ricordiamo che quest'ultimo è definito "ad 8 bit", ma in realtà il 6809 è un micro processore che ha la struttura interna a 16 bit (registri, etc.): in effetti il 6809 va inquadrato come il chip di "ponte" fra le primitive CPU ad 8 BIT e la CPU a 16 bit dell'ultima generazione, l'MC68000.

CELDIS ITALIANA SPA.
Via F.lli Gracchi 36
CINISELLO BALSAMO (MI)

Sistema di prototipaggio per le schede Z80 'MD'

Nell'ambito del programma di sviluppo della famiglia di schede a microprocessore (Z80/Z80A) 'MD' realizzate secondo il BUS standard "STD", la Mostek offre un compatto sistema di valutazione e prototipaggio denominato MDX-PROTO.

Il PROTO è un kit composto di hardware e software che consente di sviluppare progetti basati sul microprocessore Z80/Z80A e di realizzare sistemi a partire dalle schede MD-STD.
Infatti incorpora le schede CPU1 (Z80, CTC, 4K EPROM, generazione BUS-STD), DRAM8 (8K di RAM, espandibili a 32K), DEBUG (10K di ROM contenente il sistema operativo monitor, editor, assembler, loader e un'interfaccia seriale).
Il PROTO comprende inoltre una scheda prototipi, una scheda prolunga, cavi di collegamento per periferiche seriali, il cestello con 8 posti scheda e tutta la documentazione.
L'MDX-PROTO costituisce quindi un sistema minimo di sviluppo per Z80 con software in ROM; gli utenti che richiedono memoria di massa possono passare al sistema di sviluppo universale Mostek, il SYS 80FT.

MOSTEK ITALIA - MILANO

PERIFERICHE

Serie 1000: plottes "J & J Instr. Ltd"

La J & J presenta la serie 1000 dei plotters a 1, 2 e 3 penne, che così vanno a completare la gamma dei semplici registratori già prodotta dalla stessa casa.
Tali plotters incorporano l'avanzamento della carta e lo sweep della base dei tempi, ma controllati al quarzo.
Le caratteristiche salienti per l'utente sono la possibilità di alloggiare la carta sotto il piano di scrittura per risparmiare spazio; un piano di scrittura scorrevole per facilitare l'accesso alle parti elettroniche per servicing; caricamento automatico della carta; sprocket antislittamento per l'avanzamento della carta.
Inoltre si nota la presenza di logica completa per interfacciare computers, cambio automatico della carta, alzapenna, etc., e la possibilità, tramite un completo pannello di comando, di agire sul convertitore F/V, sul calibratore per tutti gli ingressi, sul comando di verifica di zero reale senza dover sconnettere l'ingresso.
Tutti gli ingressi sono adatti per segnali ac o dc, e i terminali di ingresso sono localizzati nella base per disimpegnare i comandi.
Le prestazioni ottenibili sono una precisione dell'uno per mille, una velocità di scrittura di 1000 mm/sec., offset calibrato fino al 1000%, convertitore frequenza/tensione; precisione nel trascinamento della carta dello 0,01%. È data inoltre la possibilità di inserire il plotter su rack.
Sono disponibili versioni OEM.

ELIS srl
Via Nitti 18, ROMA

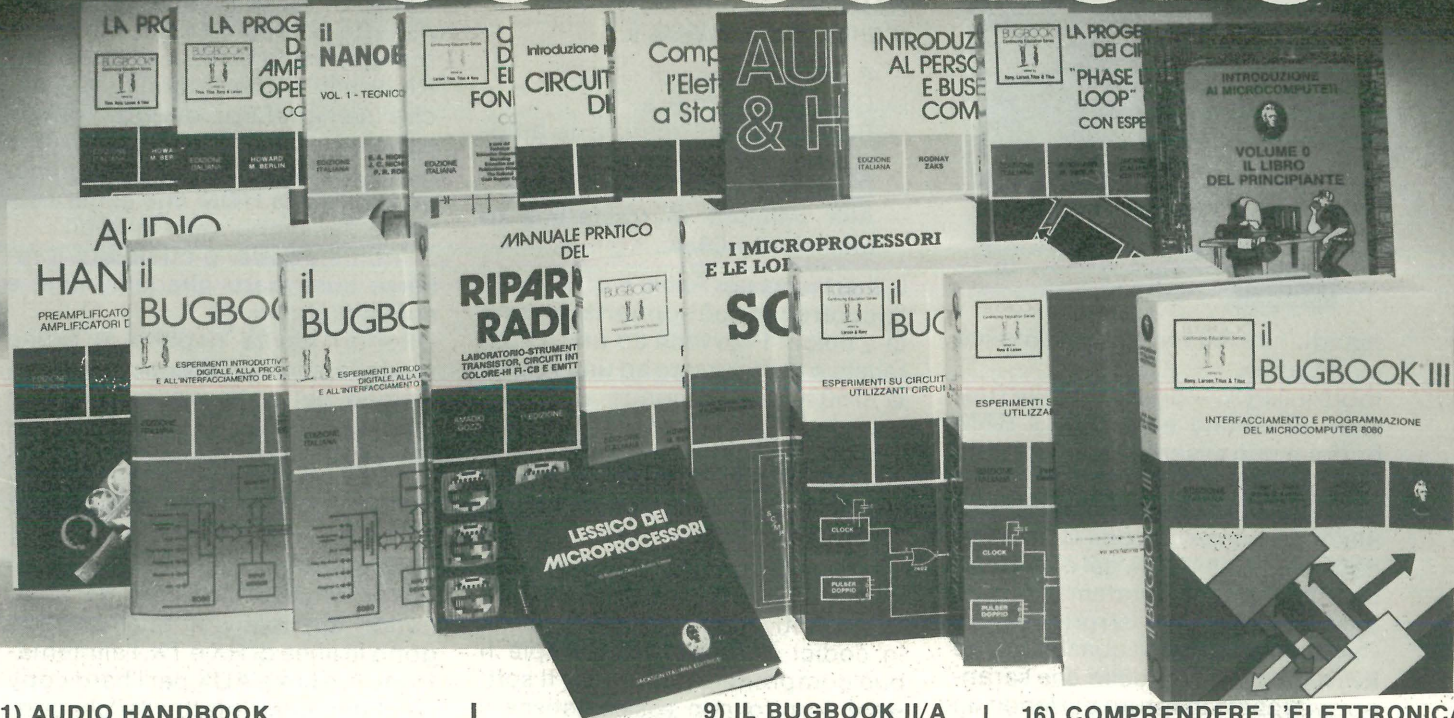
SA1000: il più economico Winchester

Sono disponibili i nuovi dischi fissi a tecnologia Winchester della Shugart. Interessante è l'unità SA1000.
La capacità del disco fisso consente di allargare le applicazioni dei sistemi a microproccessore, fornendo una periferica per la memorizzazione dati altamente efficente sia in termini di capacità e di affidabilità che di costo.
Le caratteristiche dell'unità SA1000 sono:
— capacità 5 e 10 Megabytes
— dimensioni esattamente uguali a quelle di un floppy-disk
— interfaccia e alimentazione simili a quelle di un floppy-disk.
L'unità SA1000 può essere inserita in sistemi gestionali in coppia con un floppy. È disponibile inoltre un controller-formatter per la gestione del disco fisso più floppy a doppia testa, doppia densità.

TELCOM
Via Civitali, 75, MILANO

Programmatore di memorie

L'EPROM programmer & Z80 computer Pecker-I PDW—5000 della Toyo Telesonics visualizza in modo chiaro, con un display a LED a 7 segmenti a 16 cifre, i vari modi di funzionamento (quali comandi simbolici, nomi degli indirizzi e dei dati della ROM scelti). Nonostante sia contenuto in un cabinet portatile di soli 48 mm, il programmatore è dotato di un alimentatore incorporato.
La tastiera, divisa in due sezioni per i tasti dei dati e i tasti dei co-

i "Best-Sellers"

1) AUDIO HANDBOOK
Manuale di progettazione audio con progetti completi.
L. 9.550 (Abb. L. 8.550)

2) IL BUGBOOK V
Esperimenti introduttivi all'elettronica digitale alla programmazione e all'interfacciamento del microprocessore 8080 A.
L. 19.000 (Abb. L. 17.100)

3) IL BUGBOOK VI
Completa la trattazione del Bugbook V.
L. 19.000 (Abb. L. 17.100)

4) MANUALE PRATICO DEL RIPARATORE RADIO-TV
Il libro scritto da un riparatore per i riparatori.
L. 18.500 (Abb. L. 16.650)

5) IL TIMER 555
Oltre 100 circuiti pratici e numerosi esperimenti.
L. 8.600 (Abb. L. 7.740)

6) SC/MP
Applicazioni e programmi sul microprocessore SC/MP.
L. 9.500 (Abb. L. 8.550)

7) IL BUGBOOK I
Esperimenti su circuiti logici e di memoria utilizzanti circuiti integrati TTL.
L. 18.000 (Abb. L. 16.200)

8) IL BUGBOOK II
Completa la trattazione del Bugbook I.
L. 18.000 (Abb. L. 16.200)

9) IL BUGBOOK II/A
Esperimenti di interfacciamento e trasmissione dati utilizzanti il ricevitore trasmettitore universale asincrono (UART) e il Loop di corrente a 20 mA.
L. 4.500 (Abb. L. 4.050)

10) IL BUGBOOK III
Interfacciamento e programmazione del microcomputer 8080 A.
L. 19.000 (Abb. L. 17.100)

11) LA PROGETTAZIONE DEI FILTRI ATTIVI
Tutto ciò che è necessario sapere sui filtri attivi.
L. 15.000 (Abb. L. 13.500)

12) LA PROGETTAZIONE DEGLI AMPLIFICATORI OPERAZIONALI
Tutto ciò che è necessario sapere sugli OP-AMP.
L. 15.000 (Abb. L. 13.500)

13) IL NANOBOOK - 80 - VOL. 1
Tecniche di programmazione.
L. 15.000 (Abb. L. 13.500)

14) CORSO DI ELETTRONICA FONDAMENTALE
Testo ormai adottato nelle scuole per il suo alto valore didattico. Per capire finalmente l'elettronica dalla teoria atomica ai circuiti integrati attraverso una esposizione comprensibile a tutti. Esperimenti e test completano la trattazione.
L. 15.000 (Abb. L. 13.500)

15) INTRODUZIONE PRATICA ALL'IMPIEGO DEI CI DIGITALI
Consente un rapido apprendimento dei circuiti integrati.
L. 7.000 (Abb. L. 6.300)

16) COMPRENDERE L'ELETTRONICA A STATO SOLIDO
Un corso autodidattico in 12 lezioni per comprendere tutti i semiconduttori e come questi funzionano insieme in sistemi elettronici.
L. 14.000 (Abb. L. 12.600)

17) AUDIO & HI-FI
Una preziosa guida per chi vuole conoscere tutto sull'hi-fi
L. 6.000 (Abb. L. 5.400)

18) INTRODUZIONE AL PERSONAL & BUSINESS COMPUTING
Un'introduzione esauriente e semplice al mondo affascinante del microcomputer.
L. 14.000 (Abb. L. 12.600)

19) LA PROGETTAZIONE DEI CIRCUITI PLL
Tutto ciò che è necessario sapere sui circuiti "Phase Locked Loop" (PLL).
L. 14.000 (Abb. L. 12.600)

20) INTRODUZIONE AI MICROCOMPUTER VOL. 0 IL LIBRO DEL PRINCIPIANTE
Un corso per coloro che non sanno niente (o quasi) sui calcolatori e gli elaboratori.
L. 14.000 (Abb. L. 12.600)

21) LESSICO DEI MICROPROCESSORI
Un pratico riferimento a tutti coloro che lavorano nel campo dei microcalcolatori o che ad esso sono interessati.
L. 3.500 (L. 3.150)

CEDOLA DI COMMISSIONE LIBRARIA
da inviare a Jackson Italiana Editrice srl - Piazzale Massari, 22 - 20125 Milano

Nome ______________________

Cognome ______________________

Via ______________________ N. ______

Città ______________________ Cap ______

Codice Fiscale (indispensabile per le aziende) ______________________

Data ____________ Firma ______________________

SCONTO 10% AGLI ABBONATI

Inviatemi i seguenti volumi:

☐ Pagherò al postino l'importo indicato più spese di spedizione

☐ Allego assegno n° ______________________

di L. ______________________ (in questo caso la spedizione è gratuita)

☐ **Abbonato** ☐ **Non abbonato**

Barrare i numeri che interessano

1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21

mandi, funziona in modo semplice e affidabile.
Inoltre il PDW—5000 ha molte funzioni di editing attraverso il RAM buffer e può essere usato anche da non specialisti.
Un piccolo emettitore sonoro fornisce la conferma dell'inserzione dati attraverso i tasti, del completamento della programmazione e della eventualità di errori.
Può programmare quasi tutte le EPROM attuali e quelle che saranno sviluppate in futuro partendo da 4K bit fino a 32 Kbit. Il cuore del programmatore è costituito da una CPU Z80 e 16K di RAM dinamiche.
Il Pecker-I è poi facilmente espandibile, aggiungendo una scheda nel contenitore.

Toyo Telesonic
ELIND - CERNUSCO S/N

PERSONAL COMPUTERS

Nuove periferiche per l'Apple II

Sono disponibili due nuove periferiche per il personal computer Apple II: il Corvus 11AP, un hard disk da 10 megabytes, e la tavoletta grafica per l'introduzione di disegni ad alta risoluzione usando una "penna elettronica" su di una tavoletta magnetica.
Il Corvus 11AP è una periferica intelligente che potenzia notevolmente le capacità di memoria di massa dell'Apple II, mantenendo completa compatibilità con l'hardware ed il software esistenti.
Il sistema è composto dal driver IMI 7710 Winchester, dal controller intelligente Corvus, dal sistema di alimentazione e da un modulo intelligente per l'Apple II, consistente in una scheda di interfaccia e nel software ad essa associato.
Il controller Corvus è basato su un microprocessore Z80 con 16K di RAM.
Il firmware provvede a:

— Bufferizzazione di settori
— Controllo scrittura
— Recovery con ripetizione automatica
— Controllo errore CRC
— Alta velocità di trasferimento utilizzando DMA.

L'interfaccia per Apple II e l'associato controller può sopportare 4 unità a disco. Provvede ad un I/O parallelo e bufferizzato ad una velocità di 50 Kbytes/secondo. Il software è contenuto in due ROM presenti sulla stessa interfaccia.
L'unità ad hard-disk è compatibile sia col DOS 3.2 sia con il sistema Pascal.
La tavoletta grafica è una periferica che converte la posizione ed i movimenti di una speciale penna in codici numerici che l'Apple II può comprendere ed usare. Il software fornito con essa gestisce il disegno sul video grafico ad alta risoluzione, usando le informazioni fornite dalla tavoletta. Questi programmi trasformano l'Apple II in una tavolozza per artisti, in un tecnigrafo, oppure in una semplice lavagna elettronica.
Con la tavoletta grafica e gli appositi programmi è possibile disegnare a mano libera sul video collegato all'Apple II, oppure tracciare linee rette, rettangoli, creare cornici o singoli punti.
La figura creata può essere facilmente salvata su disco e richiamata ogni qualvolta lo si desideri. È inoltre possibile usare l'Apple II per calcolare aree, distanze fra disegni o linee disegnate sulla tavoletta e, infine, cambiare scala al disegno.
L'uso della tavoletta grafica non implica una approfondita conoscienza dell'Apple II, e quindi è alla portata di chiunque.

IRET - REGGIO EMILIA

Nuovo terminale video semigrafico della SYNERTEK

La SYNERTEK ha introdotto da poco tempo la versione 80 dell'ormai famoso terminale video KTM-2 (Keyboard Terminal Module).
Le caratteristiche salienti sono identiche a quelle del KTM-2; però notevoli modifiche sono state fatte per assicurare un video senza problemi con formato di ben 80 caratteri su 24 righe: in pratica il doppio rispetto al vecchio KTM-2, che comunque resta tuttora disponibile.
Ricordiamo che il terminale è anche semigrafico, ovvero, tramite appositi comandi di tastiera, con degli escape si può agevolmente accedere alla ROM che genera sul video dei simboli semigrafici.
È inoltre possibile effettuare il reverse sulle parti alfanumeriche e sui semigrafici.
Ulteriori novità rispetto al primo modello sono una particolare precisione del software di controllo che gestisce la rappresentazione dei caratteri sul video; molti problemi - relativi soprattutto al funzionamento a 50 Hz - sono stati eliminati.
Inoltre una modifica è stata ottimamente effettuata sul set dei connettori applicativi ai quali convergono le linee di RX e TX, l'alimentazione e le linee AUX per l'hard copy su stampante. Infatti i vecchi connettori passo 3.96 sono stati sostituiti dai più razionali Cannon a vaschetta, che hanno un'estetica del resto migliore.
Il microswitch posto sulla scheda permette all'utente di selezionare con assoluta semplicità fino a 9 baud rates diverse, ed inoltre è data la possibilità di selezionare il video, interlacciato o meno, il tipo di parità, ed il sincronismo a 50 o 60 Hz.
Ricordiamo che il KTM-2 (in entrambe le versioni) porta già sulla stessa scheda la tastiera Full ASCII.
Con la nuova versione a 24 x 80, il terminale KTM-2 può essere inquadrato fra i terminali video più competitivi e professionali: un'ottima scelta per l'hobbysta-consumer (sopratutto se possiede il SYM) ed una scelta quasi obbligata per chi - potendo disporre di un sistema di sviluppo per lavoro ed esigenze di progettazione - ha bisogno di un terminale dalle indubbie caratteristiche ottimali.

COMPREL srl - MILANO
SKYLAB srl - MILANO

Novità nella famiglia Minimicro

Sette nuove schede Eurocard "Minimicro" saranno presentate al BIAS '80 da due ditte Bresciane. La

EC Elettronica, che ha introdotto nel '79 lo standard Minimicro, ha sviluppato una scheda universale di I/O denominata VIO (Versatile Input/Output). Specifica per interfacce industriali in presenza di forti disturbi, è componibile con moduli di I/O isolati otticamente fino ad un totale di 20 linee; le uscite sono floating, portano 1A continuo, sono adatte per carichi induttivi e protette contro sovraccarichi e cortocircuiti.
Sempre la EC Elettronica presenta la Minimicro CRT, una memoria video a formato programmabile che può funzionare sincrona con il clock di sistema, in modo da non avere mai nè perdita di tempo della CPU nè disturbi sul video. Con l'aggiunta della CRT PAL si può avere una serie di attributi per ogni singolo carattere: reverse, lampeggio, grafico, colore (8 colori in standard PAL o NTSC o scala di 8 grigi), fondo complementato, carattere memorizzato ma non visibile; varie altre possibilità riguardano inoltre il colore di fondo e, in generale, l'intero schermo. La scheda appare come un'area di memoria di 4K.
La Computerjob ha realizzato la Minimicro RAM/VIA, una memoria statica da 8K che contiene anche una VIA (20 linee di I/O, 2 timers a 16 bit, uno shift register); il connettore applicativo della scheda è compatibile con quello della Minimicro 65.
La Computerjob presenta anche la SPI (Serial Parallel Interface), che monta tre VIA (60 linee TTL) ed una ACIA 6551 per comunicazione seriale con l'intefaccia TTL, loop 20 mA ed RS-232. La scheda occupa in memoria soltanto 64 bytes.
Tutte le nuove schede sono nate per applicazioni industriali, ma sono naturalmente utilizzabili come espansioni per KIM-1, SYM-1, AIM 65. A questo scopo la Computerjob presenta la Sisterboard una interfaccia universale tra uno dei sistemi citati, e tre schede della famiglia Minimicro; il tutto può essere montato sotto o a fianco del sistema stesso.
Dalla EC, inoltre è stato sviluppato il BIS (Bus Interface System), che permette di studiare progetti con le schede Minimicro montate in rack usando come sistema di sviluppo il KIM-1, il SYM-1, l'AIM 65, il SYSTEM 65, l'EXORciser o l'EXORset (ricordiamo che la famiglia Minimicro è compatibile con tutte le CPU della serie 6800).

EC ELETTRONICA
Via G. Gaggia, 22, BRESCIA
COMPUTERJOB ELETTRONICA
Via Molinari, 20, BRESCIA

SOFTWARE

Biblioteca per microcomputer

La Siemens mette a disposizione la biblioteca matematico-tecnica per i sistemi a microcomputer di tipo SAB 8080/8085, rendendo così accessibile quasi tutto il suo know how in materia di programmazione al fine di risolvere i più comuni problemi matematici. La biblioteca è memorizzata su un unico floppy-disk e un manuale dettagliato ne facilita l'accesso: si presuppone tuttavia il possesso di un sistema di sviluppo SME.
La nuova biblioteca viene messa a disposizione sotto la denominazione FSL 85 (Fundamental Support Library) ed è classificata secondo i seguenti 9 settori: la macchina FSL, l'elaborazione string, l'aritmetica dei numeri interi, decimali e del floating point, la conversione dei numeri e le operazioni di input-output, funzioni statistiche, algoritmo PID nonchè funzioni matematiche elementari.
La biblioteca offre l'aritmetica dei numeri interi per operandi senza segno (8/16 bit) e con segno (8/16/32 bit). È possibile fare uso dell'aritmetica del floating point con precisione semplice, doppia e ampliata. Routines di conversione permettono la rappresentazione interna di diversi formati. Un pacchetto string elabora successioni di segni (strings) ed effettua il loro input/output. Per l'aritmetica decimale con segno possono venir elaborati operandi fino a 32 cifre di lunghezza.
Gran parte dei 250 programmi si riferiscono alla elaborazione di processi. Vi sono delle funzioni per compiti di regolazione (algoritmo PID) e numerose routines di statistica.
Per la determinazione delle funzioni trigonometriche e trascendenti vi è a disposizione un pacchetto di sequenze matematiche elementari che comprendono le funzioni trigonometriche, esponenziali, nonchè le funzioni inverse. 1 calcoli possono venir effettuati rispettivamente con precisione semplice o doppia.
La biblioteca FSL 85 permette agli utenti un minor impiego di tempo e di mezzi finanziari. I programmi rendono inoltre possibile numerosissime combinazioni nel caso in cui la soluzione del problema esiga diversi tipi di calcolo.

SIEMENS - MILANO

DOS per i microcomputers General Processor

È finalmente disponibile il tanto atteso Disk Operating System per i microcomputers della General Processor. Il nuovo DOS permette all'utente di usare nel migliore dei modi le caratteristiche della macchina, garantendo la possibilità di creare dei files identificati soltanto da un nome simbolico. Il DOS è stato progettato dalla Digital Research di Pacific Groove (CA, USA), cui competono tutti i diritti di copyright.
La documentazione sul DOS è tutta in lingua inglese ed è composta da 5 sezioni:
— An introduction to CP/M features and facilities
— A context Editor for CP/M
— CP/M Assembler User's Guide
— CP/M Dynamic Debugging Tool
— CP/M Interface Guide
Riportiamo alcune caratteristiche tecniche del DOS:
— STAT: per ottenere informazioni sullo stato della macchina (occupazione, lunghezza dei files, etc.) e per modificare l'assegnazione dei dispositivi di input/output
— ASM: assembler
— LOAD
— DDT: debug simbolico con assembler/disassembler incorporato
— PIP: programma per lo scambio di files tra dischi o tra periferiche diverse
— ED: editor
— SYSGEN: per duplicare dischi di sistema
— SUBMIT: per creare programmi di comandi susseguenti per il DOS. Ideale per le elaborazioni in batch
— DUMP: per la stampa esadecimale dei files.

GENERAL PROCESSOR
FIRENZE

4208: IL CONCETTO DI REGISTRAZIONE DATI A PROVA DI ERRORE

- Basata su microprocessore
- Compatibilità ECMA 34 e SILENT 700®
- Memoria con doppio buffer da 256 caratteri
- Batteria tampone
- Compatibilità con perforatore FACIT 4070
- Formattazione automatica con controllo CRC
- Varie interfacce: Facit Universal, Facit SP1, V.24/RS-232-C e Custom I/0 Board.

MILANO - Via Toffetti, 2 - Tel. 56.94.245-6-7-8-9
ROMA - Via D. Cerquetti, 39 - Tel. 53.46.956
TORINO - Corso G. Lanza, 88 - Tel. 65.15.48
BOLOGNA - Via Boldrini, 6 - Tel. 55.83.11

SEMPRE QUALCOSA IN PIÙ NEL CAMPO DELLE PERIFERICHE

FEEDBACK

L'elettronica è grande, piena di cose da sapere, semplici e complesse. Se a questo aggiungete che in particolare l'elettronica digitale è una scienza giovane, le perdonerete facilmente la colpa di non essere ancora adeguatamente insegnata in maniera organica e completa. Non che non la si insegni, ma appena adesso fanno capolino degli istututi che insegnano *specificatamente* l'elettronica, anziché occuparsene come di un complemento necessario, ma marginale.

Ne segue purtroppo che in questo campo siamo un pò tutti degli autodidatti; quindi, chi più, chi meno, abbiamo tutti delle lacune grandi o piccole nella nostra preparazione elettronica, e ce ne vergognamo come ladri. Lo specialista dei computers avrà dei problemi con i phase locked loops, il mago dei dischi magnetici nell'intimità si chiederà che diavolo sia una fase di fetch, eccetera; e ciascuno farà del suo meglio per nasconder la propria ignoranza specifica, di solito annuendo con aria intelligente la sua mente brancola nel buio.

Per ovviare a questo nell'ambito di FEEDBACK risponderemo alle domande imbarazzate ed imbarazzanti di natura eminentemente tecnica e scientifica, che avrete il pudore di farci pervenire anche in forma anonima: un po' come "tutto quello che avreste voluto sapere sull'elettronica ma non avete mai osato chiedere", una maniera spiccia ma esauriente di tappare le falle della propria preparazione elettronica.

Se l'idea vi piace, scriveteci le vostre perplessità elettroniche; se non vi piace, scriveteci ugualmente per dircelo: lo spazio è troppo prezioso per sprecarlo in qualche cosa che non vi va.

Fabio Fumi

Interprete BASIC su EPROM

Sto entrando anche io (a piccoli passi, con tanti sacrifici, tanto volontà e un tantino di ... ignoranza) nel mondo dei micro.

Sono in possesso di scheda CPU (Z80) con relativa RAM e monitor esadecimale in ROM, interfaccia video e tastiera alfanumerica. Ancora poco, vero? Comunque!

Il mio problema è questo:
programmare in esadecimale è esasperante e non posso almeno per il momento acquistare un'interfaccia per cassette con relativo interprete BASIC.

Posso disporre, anche se relativamente, di un programmatore di PROM-EPROM, quindi sono agevolato da una parte.

Ora arrivo al dunque: vorrei sostituire la ROM (con il monitor esadecimale con una PROM o EPROM programmata con interprete BASIC, ed a questo proposito cade proprio a pallino l'articolo iniziato sul n. 4 di BIT dal titolo "Interprete BASIC in 8080".

Fin qui OK, ma non so (e qui sta la mia ignoranza) come programmare queste benedette PROM cioè come devo scrivere l'interprete, cosa devo scrivere in memoria!

Faccio un esempio:

002C CD8600 R 58 CALL IGNBLK IGNORA I BLANK NON SIGNIFICATIVI

Questa è una linea del programma; come devo scrivere la memoria, quale della linea devo scrivere?

Per favore mi faccia un esempio con la linea di qui sopra; so che poi sarà come sempre l'uovo di Colombo, ma da solo non ci arrivo.

La ringrazio fin d'ora porgendoLe le mie scuse e faccio i miei più vivi complimenti per la Vostra e Nostra Rivista, che è eccellente.

"Groviglio", Cagliari

Perdonami se per ragioni di difficoltà grafiche ho chiesto di tradurre con Groviglio quell'interessante ideogramma con cui ti firmi e che a buon diritto mi sento di interpretare come uno pseudonimo. Niente di male, beninteso, ma debbo toglierti l'illusione che la tua firma sia leggibile.

Ti pare poco quello che hai? Sciagurato! È un sistemino niente male, forse appena appena scarno, ma sufficiente per lavorarci e tirarne fuori delle buone soddisfazioni; penso che parecchi dei nostri lettori te lo invidierebbero, tienilo d'occhio.

Te lo invidieranno ancora di più quando, superato il tuo piccolo uovo di Colombo, sarà programmabile in BASIC. Vediamo un pò il problema.

IL listing riportato sul numero 4 e 5 di BIT è in Assembler 8080 riallocabile, come tutti gli Assembler, ha le sue regole e le sue convenzioni, se non le conosci, hai un bel dibatterti ch ele tue possibilità di uscirne vincitore non migliorano. Quindi non te la prendere se non ci sei arrivato con le tue forze, sei giustificato.

Leggiamo la tua riga-esempio e vediamo il significato dei singoli campi:
002C è l'indirizzo di PROM (o EPROM, come preferisci) a cui comincia l'istruzione contenuta nel resto della riga; se l'istruzione occupa un singolo byte, è anche l'unico indirizzo di esso, mentre, come in questo caso, se l'istruzione occupa tre bytes, 002C significa che la tua istruzione di CALL occuperà le voci 002C, 002D, 002E. IMPORTANTE: non si tratta di un indirizzo assoluto, ma relativo, vale a dire che, se in qualche maniera ti fa comodo che il programma incominci da un indirizzo diverso da 0000 (e nel tuo caso è così, perchè non vedo la ragione per cui dovresti sacrificare il pur scomodissimo monitor esadecimale), a questo indirizzo devi aggiungere quello della prima voce in cui effettivamente si troverà l'interpreter. Per fare un esempio sciocco, se il tuo interpreter dovrà trovarsi dalla voce 1200 alla 1800 (spero di non aver sbagliato il conto, in hex mi succede spesso), la prima voce dell'istruzione di cui stiamo parlando si dovrà trovare all'indirizzo 122C.
È una sciocchezza, ma ci tornerà utile nel seguito.

CD8600 va letto CD 86 00 ed è il codice oggetto dell'istruzione: se vai a vedere un manuale 8080 o Z80, troverai che CD è il codice che corrisponde all'istruzione CALL, seguito da due bytes che specificano l'indirizzo della subroutine cui bisogna accedere. Nota che l'indirizzo specificato in questi due bytes va letto alla rovescia, tanto per semplificare le cose; in particolare vedrai che nel listing la subroutine IGNBLK si trova all'indirizzo relativo 0086.

R significa che il codice oggetto di cui abbiamo appena parlato va rivisto nel caso tu desideri riallocare il programma, ossia farlo cominciare altrove che a zero. Nell'esempio di prima, l'indirizzo di IGNBLK diventerebbe 1286 (CD ovviamente non si tocca!) ; i tre bytes che compongono il codice diverrebbero quindi CD 86 12. Come scrivi il tuo interpreter, ricorda di modificare tutti i codici che hanno la R, gli altri vanno bene così quindi lasciali stare.

58 è il numero (decimale) della linea di istruzione, fatto che riguarda solo chi è in possesso del programma source originario e lo vuole modificare usando un editor; a te, credo non serve.

Il resto della riga è composto dal mnemonico dell'istruzione (giusto per sapere cosa vuol dire senza ricordare tutti i codici numerici a memoria) e dal commento, fondamentale solo per capire ed eventualmente modificare il source.

In definitiva:
La parte di riga che ti serve comprende i soli primi due campi, entrambi numerici.
Il primo è l'indirizzo, che va corretto secondo l'indirizzo iniziale.
Il secondo è l'insieme delle voci hex che

compongono l'istruzione; se c'è una R bisogna ricorreggere l'indirizzo rappresentato dagli ultimi due bytes, secondo i soliti criteri.

A titolo di esempio ti do il contenuto hex di un paio di voci intorno alla zona di programma che hai usato per esempio, sempre ammettendo che l'interpreter debba cominciare alla voce 1200. Nota che gli indirizzi che ti do sono relativi alla EPROM che programmerai (i bits più significativi riguardano i decorders di memoria, non la EPROM, che usa solo gli ultimi dieci (se 2708) o undici (se 2716).

INDIRIZZO	*CONTENUTO*
0029	CD
002A	DD
002B	12
002C	CD
002D	86
002E	12
002F	7C
0030	B5

eccetera

Se ti dovessi chiedere che cosa sia un campo alfanumerico ulteriore, che nell'esempio che hai citato non compare e che sta fra il numero dello statement ed il codice mnemonico, è il label (etichetta). Serve a riferirsi a quello statement (per esempio, a saltarci dentro) senza chiamarlo per numero, fatto che spesso è scomodo; ma è utile solo quando si lavora con l'Assembler. Dove il campo del label comincia con punto e virgola (;) lo statement viene ignorato dall'Assembler e serve solo di commento, per tenere ordine nel listing; ai fini del contenuto di memoria questi statements NON ESISTONO.

Buona fortuna.

P.S. Scusa se ti ho raccontato cose che già sai: nell'incertezza qualche parola in più magari aiuta, qualche parola in meno ti porterebbe a maledirmi.

Complimenti e ringraziamenti

sono una studentessa della facoltà di Matematica dell'Università di Bologna.

I miei complimenti e ringraziamenti per aver ideato questa rivista.

I complimenti sono:
— per la discreta facilità con cui si legge (una rivista che tratta tali argomenti potrebbe cadere facilmente nell'incomprensibile)
— perchè nel suo genere è nuova
— e ancora per come si presenta.

I ringraziamenti: perchè mi è servita e penso proprio che mi servirà anche in futuro (mi sono abbonata proprio per questo motivo).

Più di tutti, mi interessano gli articoli di Nucleo, Software, Personal Computer: riescono bene ad ampliare le mie conoscenze (a volte anche quelli di Hardware).

E adesso due domande:
— Perchè non ampliate la parte Software cercando di trattare le tecniche che si stanno sviluppando in questo campo, e continuando con i linguaggi di programmazione (Cobol, Fortran)?
— Avete pensato ad inserire nella rivista una specie di dizionarietto dei termini che si usano? Potrebbe servire a noi lettori.

Allora vi auguro Buon lavoro anche per il futuro.

E non pensate assolumtamente di smettere.

Franca Corelli - Bologna

Una ragazza della redazione, fermamente convinta che le menti umane si dividano in due categorie, quando ha visto che finalmente ha scritto UNA DONNA ha cominciato ad irradiare felicità fino a ionizzare l'aria.

Io stesso sono piuttosto felice che BIT abbia valicato il gap fra i sessi: credo che la bontà del computer personalissimo che sta fra le orecchie non dipenda molto dall'assetto dell'hardware periferico, e mi auguro che quando quest'idea sarà tanto ovvia da essere banale il mondo sarà un tantino migliore.

Passo imbarazzato i complimenti ed i ringraziamenti al resto della redazione, che pudicamente declina, ma comincia a camminare a qualche centimetro da terra.

Quanto alle domande, che poi sono proposte: sì, la parte software è stata ampliata e intendiamo ampliarla ancora; anche per i linguaggi stiamo facendo qualche cosa, ma, te ne prego, facci grazia di quella cosa pedestre e prolissa che è il COBOL, c'è di meglio al mondo.

Sull'idea del dizionarietto ci stiamo tuttora sbranando, non che non lo vogliamo fare, anzi. Ma ci ha preso la mano e sta diventando un'opera monumentale, ci stiamo accoltellando fra integralisti e riduzionisti.

Ti ringrazio e giuro che, se riscrivi, sarò più sollecito nel rispondere.

Alcune proposte

Io sono uno di quelli che hanno disobbedito all'ordine di non leggere. E quello che ho letto mi è piaciuto. Così come mi piace BIT.

Le parti che più interessano sono gli articoli come "Che cosa possono fare e che cosa non possono fare le macchine"; "Monitoraggio di pazienti in unità con microcomputer", per reale interesse personale. Considero molto interessanti pure i programmi in Basic pubblicati, ma vi prego, non limitatevi ai giochi!

Sono certo che la pubblicazione di una varietà di programmi seri per le più disparate applicazioni non danneggerebbe troppo l'industria del software, in quanto che la maggior parte dei suoni introiti sono per programmi particolari la conseguente svalutazione del mercato dei programmi, diciamo, "low-end", inciterebbe un più rapido sviluppo di programmi di uso generale più potenti, e più adattabili.

Molto utili secondo me, gli articoli sul software. Ricordatevi però che i principianti esistono ancora!

Bene la pubblicità, abbondante come dovrebbe essere per informare davvero.

Ho una proposta di avanzare. La carta di Bit è bellissima. Troppo.

Bit secondo me dovrebbe essere anche per gli hobbisti, sperimentatori, privati, appassionati... e col diffondersi del microcomputer anche per le massaie e simili. Per queste categorie la qualità della carta mi sembra eccessiva.

Utili e interressanti sono Newsletter, Nucleo, Accesso casuale.

Un'idea potrebbe essere quella di lasciare spazio ad articoli scritti dai lettori su diverse applicazioni dei microcomputers.

Altra idea potrebbe essere una pagina per gli annunci... scambi di programmi, apparati etc., ovviamente non a pagamento, se non vi rendereste antipatici.

Io consiglierei di dare più spazio alla risoluzione di problemi specifici per mezzo del computer.

Un argomento non trattato che personalmente mi interesserebbe sarebbe la Robotica.

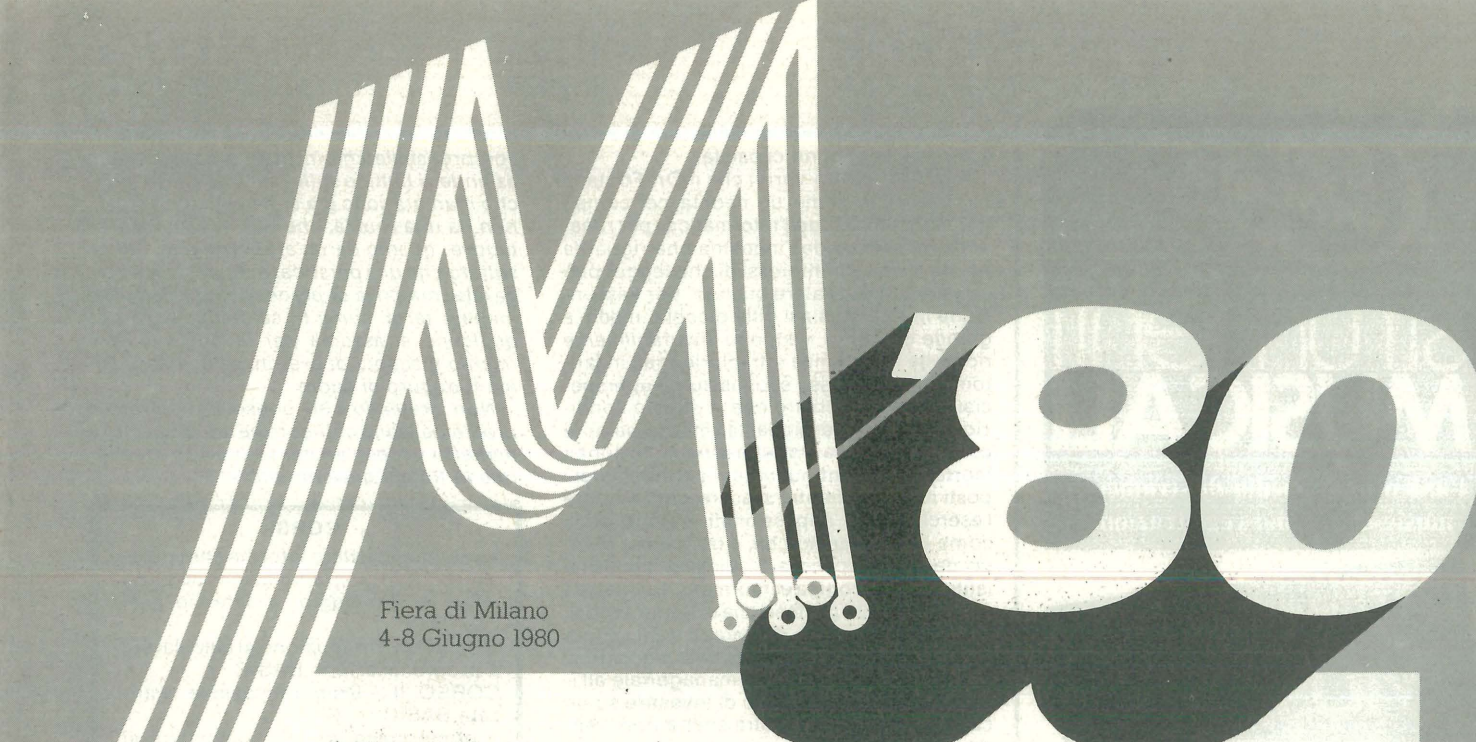
Con queso chiudo e vi saluto, promettendovi una marea di complimenti in futuro, sicuro che ascolterete le opinioni di tanti lettori che come me vi avranno scritto.

Roger Stewart, Milano

Sono d'accordo con quel che dici circa il software; quello che - debolmente - ci trattiene non è la possibilità di danneggiare le software-houses (come dici giustamente, lavorano ad un livello ben diverso), ma il rischio di perderci nella superspecializzazione e di terrorizzare i principianti. Continueremo a mantenerci in equilibrio fra questo rischio e quello opposto, di cadere nel dilettantismo.

Non sono d'accordo invece sull'opportunità di peggiorare l'estetica della rivista per risparmiare poche lire sulla carta. BIT non è rivista da buttar via dopo averla letta; vogliamo che si possa conservare in biblioteca, per consultarla un futuro, senza vergognarsene. Dopotutto quel che paghi in edicola è più il know-how ed il lavoro umano necessario a fare la rivista, che non la cellulosa laminata.

Gli articoli scritti dai lettori sono e saranno i benvenuti: questa rivista deve servire da punto di confronto fra moltissime esperienze, non da dispensatore di informazioni che cadono dall'alto. Se un giorno questi articoli saranno tanto e tanti regolari da volere uno spazio apposito, saremo felicissimi di darglielo, per ora ogni articolo valido fa caso a sé.

Se siamo simpatici o antipatici ce lo dovete dire voi, raramente una persona vede da sé i propri difetti; questa rubrica serve anche da ufficio reclami, e il fatto che finora nessuno l'abbia usata come tale ci fa sperare per il meglio. Ma ci arrabbieremmo davvero se qualcuno ci abbandonasse senza almeno dirci il perchè.

OK per la Robotica, credi che tutti quanti vediamo l'attuale situazione come un punto di passaggio dal computer gigantesco e un po' tonto al computer piccolissimo, versatile e (chissà?) umanoide nella forma e nelle prestazioni.

Per ora siamo appena agli inizi, ma non è una buona ragione per non occuparsene. Lo faremo.

Professionalità all'italiana?

Vorrei rispondere in merito all'articolo apparso su BIT n. 5, nella rubrica TRIBUNA, e firmato dal Dr. Giuseppe Fontana, in cui si polemizza su una mancata professionalità da parte degli italiani, soprattutto nel campo delle tecnologie più avanzate, addossandone la responsabilità al sistema e alla scuola in generale (guarda caso!).

Ora, mi pare che non si possa pretendere dalla scuola, a causa delle sue intriseche caratteristiche di generalità, una preparazione tale da mettere in grado lo studente (neodimplomato, o neolaureato che sia) di affrontare le complesse problematiche connesse ad una professione di livello tecnico in industrie di avanguardia; secondo me, anche auspicando il ritorno ad una scuola di "stampo antico" (inteso nel senso migliore del termine, vale a dire impegno e serietà da parte dello studente), non ci creeranno mai dei buoni tecnici, se questi tecni-

ci potenziali non vengono messi in grado di esternare le proprie capacità.

A questo punto vorrei che il Dr. Fontana mi spiegasse come un neofita nel campo dell'elettronica o dell'informatica (per rimanere nel settore dell'industria che riguarda più da vicino gli interessi di che legge) può sperare di farsi valere quando, per l'assunzione in una qulsiasi ditta piccola, media e grande del ramo, vengono *invariabilmente* richieste esperienze ultrapluriennali in fantomatiche e spesso sconosciute superspecializzazioni. È chiaro che, a queste condizioni, l'alta professionalità rimarrà sempre confinata in una ristretta èlite di fortunati mortali, che continueranno a palleggiarsi i posti migliori, mentre si accrescerà sempre l'esercito degli appassionati, ex-entusiasti come il sottoscritto che, a trent'anni, dopo un decennio speso a crogiolarsi tra libri e istituti di elettronica e informatica, continua ad esercitare la professione di commesso di negozio (e neppure di stereo HI FI!). È poi parliamo di professionalità all'italiana! Parliamo invece di mentalità manageriale all'italiana: paura del rischio di investire su un capitale umano che potrà anche non "rendere", scarsa lungimiranza nell'indivduare i potenziali "cavalli di razza" o, quanto meno, i futuri buoni tecnici, e mancanza di considerazione per tutti coloro che hanno speso anni per il conseguimento di un obiettivo in cui credevano. Vi sembra che abbia esagerato? Allora facciamo una prova; faccio questo annuncio, rivolto a tutti coloro che leggono e che possono essere interessati:

Ho trent'anni, sono diplomato in elettronica ed in possesso di un attestato di specializzazione su microcomputers, conseguito presso un importante istituto torinese e nessuna esperienza nel settore; sono interessato ad un impiego come analista e/o programmatore (ho una discreta conoscenza del linguaggio BASIC, con alcune esperienze di programmazione); c'è qualcuno disposto a mettermi alla prova?

Ti ringrazio per l'attenzione e per lo spazio che mi vorrai concedere nella tua rubrica.

Piercarlo Gennero - Torino

Capisco la tua amarezza e capisco anche le ragioni del Dr. Fontana, della cui pazienza abuso spesso per sfogarmi quando mi imbatto in qualcosa di italiano che funziona male: è un uomo molto comprensivo. Passo la tua lettera, pari pari, al Dr. Fontana (Beppe, se mi leggi batti un colpo). Ma lasciami aggiungere qualcosa in proposito.

Non credo che in Italia manchino i buoni tecnici, effettivi o potenziali, né penso che tutte le scuole e gli istituti in genere pecchino di superficialità nella preparazione degli allievi. Quel che è difficile, dal punto di vista dell'imprenditore, è distinguere il grano dalla pula. Mi spiego: di uno che usciva da una scuola, come dici tu, di stampo antico, si sapeva qualcosa: in genere era una persona capace e ragionevolmente preparata; mal che andasse, non era un'aquila, ma era almeno sufficientemente paziente e testardo da farsi venire il didietro piatto a furia di stare sui libri. Da uno che esce da una scuola che riconosce il titolo di studio come un diritto sociale (bada, non la preparazione, il titolo) non si sa nulla; dire che uno è laureato è un po' come dire che ha l'automobile, non specifica affatto che tipo di persona sia e che cosa sia capace di fare. D'altra parte assumere una persona per un periodo sufficiente a valutarla abbastanza bene significa, per ragioni sindacali, doverla tenere anche se non è adatta, a meno che non sia disposta ad andarsene spontaneamente. È comprensibile che in queste condizioni le aziende si buttino sulle persone che in qualche maniera sono già state valutate altrove; non da una scuola, che non valuta ma promuove, quanto da un'altra azienda. Col risultato che una persona, magari seria e preparata, ma priva di precedenti di lavoro, rimane a terra, come tu sai bene; salvo che qualcuno si assuma per la prima volta il rischio e scopra di aver investito bene, come ti auguro di cuore.

Non pretendo con queso di risolvere il problema, sono contento se sono riuscito a metterlo a fuoco anche solo un po'. La morale la tragga qualcun altro.

CORSI

La Homic Software informa che i prossimi corsi di programmazione avranno inizio il 19 maggio, e saranno così costituiti:

CORSO I - Introduzione al calcolatore e programmazione BASIC

CORSO II - Programmazione avanzata BASIC

Il primo corso intende rispondere ai quesiti più comuni di un principiante e cioé:

- Cosa è un calcolatore; - Come è fatto; - A che serve; - Come lavora; e comprenderà:

A) Fondamenti di un calcolatore; B) Microprocessore e microcomputer; C) Algebra di BOOLE; D) Cenni sui linguaggi (EDITOR - ASSEMBLER - BASIC - FORTRAN); E) Cenni sui sistemi operativi F) Funzionamento del calcolatore con le sue periferiche (es. stampante, nastro magnetico) G) la programmazione BASIC.

Saranno proposti esercizi ad ogni lezione

Il secondo corso, supponendo di rivolgersi a persone già esperte o che comunque abbiano eseguito il primo corso, propone le seguenti argomentazioni:

A) Tecnica della programmazione (applicata a programmi tecnico-scientifici-gestionali) con stesura di flow charts; B) Uso dei sistemi operativi; C) Uso di "utilities" (Sort/Merge Etc); D) Management file; E) Uso di ACCESSI SEQUENZIALI E RANDOM; F) Ottimazione di occupazione memoria - tempi di esecuzione - tempi operativi; G) Esecuzione per intero di un programma gestionale (piccolo magazzino).

Gli esercizi sia per il primo che per il secondo corso, saranno svolti sul calcolatore 3032 PET - COMMODORE.

La durata dei corsi sarà la seguente:

I° CORSO Inizio 19 maggio 1980 - Fine 4 giugno 1980 Lunedì - Mercoledì - Venerdì - ore 18 - 20

II° CORSO Inizio 16 giugno 1980 - Fine 2 luglio 1980 Lunedì - Mercoledì - Venerdì - ore 18 - 20

Il costo di partecipazione individuale ad ognuno dei due corsi è di L. 180.000. + I.V.A.

Il pagamento avverrà all'iscrizione. In caso di mancata partecipazione, purché ne sia data comunicazione 5 giorni prima dell'inizio della 1° lezione, sarà rimborsata la quota versata meno il 20%.

La Homic Software rimborserà l'intera quota in caso di annullamento del corso.

*della FLUKE naturalmente!

L.198.000+IVA

Se non hai ancora acquistato un Fluke mod. 8020 probabilmente è solo perché non sai che:

- Ha 3 digit e mezzo (±1999) • E' protetto anche nella scala degli Ohm fino a **500 V**
- E' protetto contro impulsi di tensione fino a **6000 V** • Può misurare 6 funzioni con 26 scale - (Vcc, Vca, mAcc, mAca, Ω, S) • Può misurare resistenze fino a **10.000 Megaohm** (sì, non è un errore di stampa) usando la funzione conduttanza con lettura in Siemens ($S = \frac{1}{OHM}$)
- Può misurare diodi e resistenze senza dissaldarli dai circuiti • Può misurare il coefficiente β dei transistori

□ Può sopportare urti e vibrazioni secondo le norme MIL-T-28800

SISTREL
SOCIETÀ ITALIANA STRUMENTI ELETTRONICI S.p.A.

Via Pelizza da Volpedo 59 - 20092 CINISELLO B.
tel. (02) 6181893 - Telex 320346
Via Giuseppe Armellini 39, 00143 ROMA. Tel. (06) 5915553 - Telex 680356
Via Cintia Parco S. Paolo 35, 80126 NAPOLI - Tel. (081) 7679700

□ Ricevere un'offerta □ La visita di un Vs. Tecnico □ Essere inseriti nel Vs. mailing list.

NOME .... COGNOME ....

VIA .... TEL. ....

CAP. .... CITTA' ....

DITTA .... REPARTO ....

mod. 8020 A